ESD-TR-84-045

(12)

Technical Report
705

Statistical Signal Models and Algorithms for Image Analysis

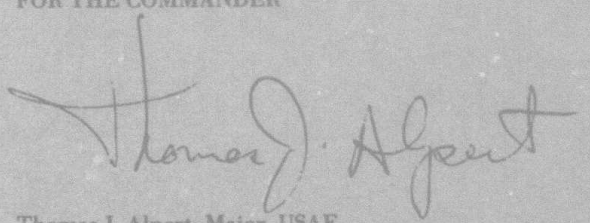
T.F. Quatieri
D.E. Dudgeon
C.W. Therrien

25 October 1984

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Defense Advanced Research Projects Agency
and the Department of the Air Force
under Electronic Systems Division Contract F19628-85-C-0002.

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored in part by the Defense Advanced Research Projects Agency and in part by the Department of the Air Force under Air Force Contract F19628-85-C-0002. (ARPA Order 3345).



MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

STATISTICAL SIGNAL MODELS AND ALGORITHMS FOR IMAGE ANALYSIS

T.F. QUATIERI

Group 24

D.E. DUDGEON

C.W. THERRIEN

Group 21

TECHNICAL REPORT 705

25 OCTOBER 1984

LEXINGTON MASSACHUSETTS

ABSTRACT

In this report, two-dimensional stochastic linear models are used in developing algorithms for image analysis such as classification, segmentation, and object detection in images characterized by textured backgrounds. These models generate two-dimensional random processes as outputs to which statistical inference procedures can naturally be applied. A common thread throughout our algorithms is the interpretation of the inference procedures in terms of linear prediction residuals. This interpretation leads to statistical tests more insightful than the original tests and makes the procedures computationally tractable. This report also examines a computational structure tailored to one of the algorithms. In particular, we describe a processor based on systolic arrays that realizes the object detection algorithm developed in the report.

TABLE OF CONTENTS

1. Introduction

This report describes an approach to image processing in which linear filtering models are used to represent portions of the image. These models generate two-dimensional random processes to which statistical inference procedures are applied. The combination of the linear models with the statistical procedures leads to practical algorithms for extracting desired information about an image. Further, although the resulting algorithms may be computationally intensive, they usually have regular structure, and, therefore, are ideal candidates for implementation with special computational architectures and structures that exploit parallelism.

This report considers three particular applications in image processing [1]: image region classification and segmentation, and object detection. Specifically, many aerial images of natural terrain can be effectively characterized as composites of textures which can be modeled by two-dimensional (2-D) random processes. Such textures may represent various types of fields, water, desert, wooded areas, and so on. The classification and segmentation of these images into regions of known type is important for a variety of applications, including image understanding and image coding and, more specifically, crop and land use data collection and cartography.In both problems, using the models cited earlier, one can develop a probability density function for the image data [2]. We use this approach to first briefly address the problem of classifying regions of known boundaries. This leads naturally into the more complex problem of image segmentation. Here, the texture model is combined with a Markov random field model to represent

the occurence of textured regions within an image. Segmentation of the image is then treated as a region estimation problem, and the resulting algorithms have a clear intuitive interpretation in terms of the output residuals of linear prediction filters [2].

The problem of detecting small regions of an image which differ from their surroundings is a third application that is developed. This problem is of considerable interest in such areas as image understanding and machine vision, and, more specifically, optical, radar and infrared image analysis, and medical diagnosis. Detection of such anomalous areas of an image is often the first step in image analysis systems which perform automatic classification of objects. In this report, we shall address the particular problem of detecting objects in aerial photographs of natural terrain such as trees, grass, and fields. The process of deciding on the presence of an object relies on a significance test which is designed to ensure Constant False Alarm Rate (CFAR) detection [3,4]. The detection algorithm derives from the fact that this significance test can be transformed to a test involving error residuals of an adaptive 2-D linear predictor and an adaptive threshold. This equivalent representation leads to an algorithm which can be computationally more tractable and also more insightful than one based on the original significance test.

The texture models used in our approach to segmentation are not unique to this report. One-dimensional ARMA (autoregressive-moving average) models for texture images were studied by McCormick and Jayaramamurthy as early as 1974 [5], and two-dimensional linear models have been extensively studied by

Pratt, Faugeras, Gaglowitz, and others [6-11]. However, the usual approaches to segmentation, even when based on these models, have centered on the analysis of texture "features" (such as moments of the co-occurrence matrix) rather than on estimation-theoretic concepts applied directly to the random fields. Where estimation-theoretic methods have been used, the results have been generally successful. Our early experiments [12] on texture images from a standard data base produced encouraging results that were further validated in comparative studies by Chen [13]. Cooper, Elliott, and their colleagues [14-16] approached the problem of boundary estimation for nontextured images in a manner similar to the way we approach region estimation for general textured images. More recently [17], Hansen and Elliott examined the problem of region estimation for images consisting of constant gray level objects in additive Gaussian white noise. The likelihood equations for this problem turn out to be similar to the equations for the maximum a posteriori estimate of textured regions described later in this paper. A significant amount of other work has been done in random field models for images. However, most of it has been oriented toward applications of spectral analysis, filtering (for image restoration and enhancement) and, to some extent, coding. Some of this work will be cited in the following sections.

The linear filtering approach to object detection is also not unique to this paper although, as before, most detection schemes are "feature dependent." Other authors have considered (adaptively) filtering background clutter to enhance object-to-background ratio [18-22]. However, the notion of close inspection of prediction filter residuals, based on significance

testing and its modification for CFAR, appears to be novel in the area of image analysis.

This object detection algorithm also serves as a good example of an algorithm amenable to a special computational structure which exploits parallelism, such as a systolic array [23]. The various steps of the algorithm of estimating a correlation matrix, solving the normal equations, and performing the prediction error filtering can be implemented on arrays most of whose cells perform the simple inner product step $c \leftarrow c + a \cdot b$ [24]. The combination of these arrays forms a highly parallel computing structure where image data flow in and object detection results flow out at similar rates.

The first two sections of the report deal with modeling and statistical inference procedures. These sections review the necessary ideas in stochastic multidimensional filtering and statistical inference and are intended to be mostly tutorial. Following that, we describe the particular topics of classification, segmentation, and object detection and derive algorithms based on the earlier statistical and modeling procedures. Finally, we focus on one particular algorithm for object detection and discuss how it could be implemented in a systolic array that pipelines the computations in a very efficient manner so that results could be carried out in real time.

2. FILTERING MODELS FOR IMAGES

To set up the discussion of image analysis through filtering models, we first develop a stochastic framework for image representation in which a pixel ordering and the notion of "past" are defined. With this as a foundation, two image models are introduced: white-noise driven and minimum-variance representations. In both models, we view an image as the output of a 2-D linear filter. In the former case, the input to the filter is white noise, while in the latter case the input is generally colored. These models are described and compared in the context of representing a 2-D random process of arbitrary positive spectral density and are shown to be equivalent under certain special conditions. While the white-noise driven model serves as a useful conceptual tool, the minimum-variance representation is more useful in practice.

2.1 Framework

We consider an image $x(n,m)$ as a sample function of a 2-D random field where n and m range over some finite rectangular grid. To begin, we require some ordering on this field and define the notion of "past" which is closely related to the notion of "causality" [25]. Specifically, for any point (n_o,m_o) in the image, we define the past to be the set of the points:

$$\{(n,m) \mid n=n_o,\ m < m_o;\ n < n_o,\ -\infty \leq m < \infty\} \qquad (1)$$

This definition is illustrated in Figure 1. As a matter of notation, if (n_1,m_1) is in the past of (n_2,m_2), we write $(n_1,m_1) < (n_2,m_2)$.

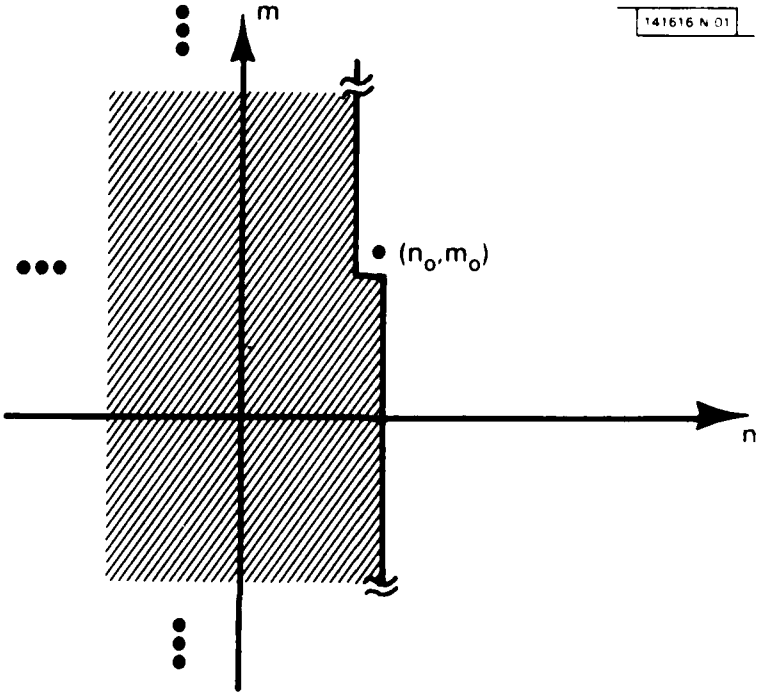


Figure 1. Definition of past.

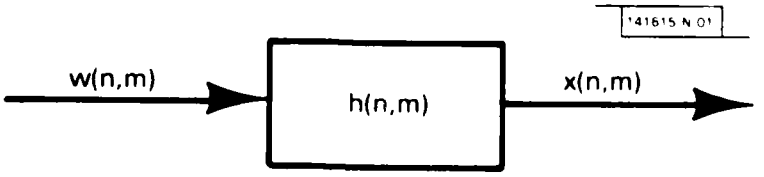


Figure 2. The white noise driven model.

If we cancatenate the columns (or rows) of $x(n,m)$ into a vector $\underline{x}$, the 2-D stochastic field is completely characterized by the probability density function $p_{\underline{x}}(\underline{x})$. The mean and covariance of the vector $\underline{x}$ can be expressed in terms of the mean and covariance functions of the random field which are defined by

$$u_x(n,m) = E[x(n,m)] \tag{2a}$$

and

$$r_x(n,m;\ell,k) = E\left[(x(n,m)-u_x(n,m))(x(n+\ell,m+k)-u_x(n+\ell,m+k))\right] \tag{2b}$$

When the field is wide-sense stationary, (2) is invariant to spatial shift; that is, u_x and r_x no longer depend on (n,m).

2.2 White Noise Driven Linear Models

A special case of the image representation of the previous section occurs when the image $x(n,m)$ is generated by a white-noise driven linear system as illustrated in Figure 2. When the white noise input $w(n,m)$ is wide-sense stationary and the linear system is shift-invariant, the resulting image is also stationary. In addition, since the system is linear, when the input $w(n,m)$ is characterized by a Gaussian probability density function the process $x(n,m)$ is also Gaussian.

One white-noise driven representation (WNDR) with which we shall be concerned is given by the linear difference equation of the form:

$$x(n,m) = \sum_{\ell,k \in M} a(\ell,k)x(n-\ell,m-k) + w(n,m) \tag{3}$$

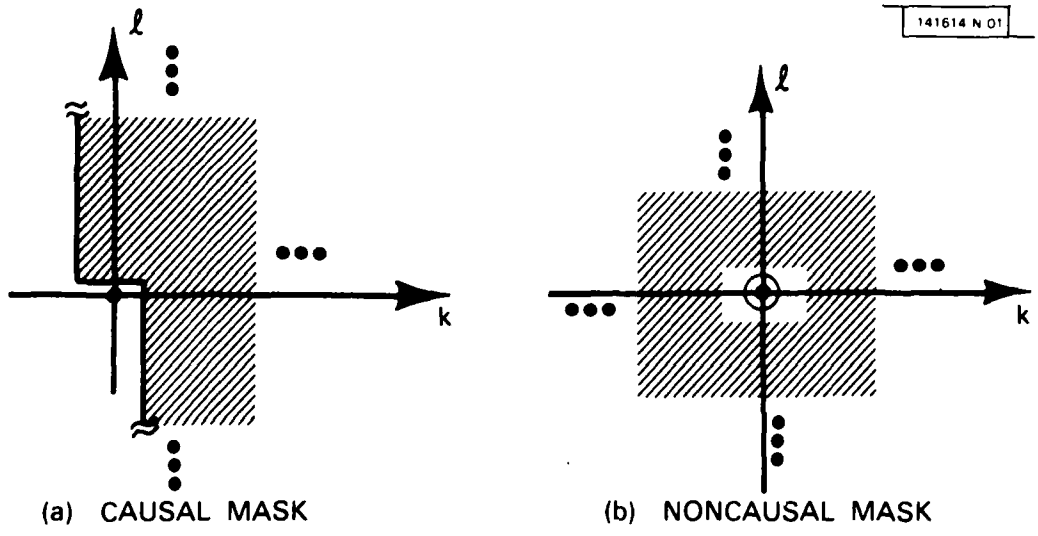


Figure 3. Autoregressive model mask supports.

where M represents the nonzero support region for filter coefficients $a(\ell,k)$ which we shall refer to as the filter coefficient mask, and where $w(n,m)$ is the white noise input with variance $(\sigma_w)^2$. Equation (3) defines an autoregressive or all-pole model (the stability of the recursion is assumed). Each image pixel of $x(n,m)$ is given by a linear combination of its surrounding samples and a single white noise sample. The coefficient mask M in this relationship may take on different shapes and, in general, may be infinite in extent. For example, when each image pixel is related to the entire set of points in its past then the shape of the mask M is of the form in Figure 3a. We shall refer to this mask geometry as a non-symmetric half plane (NSHP) [25-27] and, in this case, we say a "causal" relation exits between image pixels. More generally, the coefficient mask M may take on a geometry which is not a NSHP as illustrated in Figure 3b.

In terms of the second-order mean and covariance statistics, the WNDR with a NSHP coefficient mask is sufficiently general to characterize any 2-D random process with a positive definite covariance function [25,28]. This is true regardless of whether the actual generation process is a WNDR. The sufficiency of the NSHP mask can be seen through the equivalent second-order spectral density characterization of a random process. The spectral density function is formally defined as the inverse 2-D Fourier transform of the covariance function $r_x(\ell,k)$ or equivalently as the 2-D z-transform of $r_x(\ell,k)$, denoted by $S_x(z_1, z_2)$:

$$S_x(z_1, z_2) \triangleq \sum_{\ell,k} r_x(\ell,k)\, z_1^{-\ell} z_2^{-k} \tag{4}$$

evaluated for $|z_1| = |z_2| = 1$.

Suppose now that an arbitrary random process x(n,m) is characterized by the density function $S_x(\exp[ju],\exp[jv])$. Then, if $S_x(\exp[ju],\exp[jv])$ is strictly positive for all (u,v), it can be shown that the 2-D z-transform, $S_x(z_1,z_2)$, can be written in factored form as

$$S_x(z_1, z_2) = (\sigma_w)^2 / A(z_1, z_2) A^*(z_1, z_2) \tag{5a}$$

where we have

$$A^{-1}(z_1, z_2) = (1 - \sum_{\ell,k \in M} a(\ell,k)\, z_1^{-\ell} z_2^{-k})^{-1} \tag{5b}$$

and where the coefficient mask M is a NSHP and generally infinite in extent [26]. Furthermore, the zeros of $A(z_1, z_2)$ fall within the region $|z_1| < 1$ and $|z_2| < 1$ of the z-plane, and therefore $A(z_1, z_2)$ represents a stable 2-D polynominal. Equation (5b) can be immediately recognized as the system function of a filter corresponding to the difference equation (3). Consequently, we have shown that any 2-D random process with a positive definite covariance function can be represented by a stable autoregressive process with a NSHP support. Specifically, suppose a random process x(n,m) is characterized by a positive spectral density $S_x(\exp[ju],\exp[jv])$. Then a stable autoregressive representation (3) can be found with a white noise input of variance $(\sigma_w)^2$ and with coefficients a (ℓ,k) having a NSHP mask

geometry. This representation exists, regardless of the underlying process which generated x(n,m).

A highly useful interpretation of the function $A(z_1, z_2)$ is that of an inverse filter; i.e., it can be seen from (5) that applying x(n,m) as the input to a filter with transfer function $A(z_1, z_2)$ results in a white noise residual with variance $(\sigma_w)^2$. This inverse filtering or whitening process is particualrly important in the applications of image segmentation and object detection.

In practice, the filter parameters $a(\ell,k)$ must be determined from the available covariance function or the equivalent spectral density. From (5a), we see that obtaining this parameter set requires a 2-D spectral factorization which can be accomplished through a homomorphic transformation [26]. One practical problem in this computation involves the extent of $a(\ell,k)$. As noted, the resulting autoregressive model mask M for an arbitrary spectral density $S_x(\exp[ju],\exp[jv])$ is generally infinitely large. In image processing applications, however, it is practical to use only a small number of coefficients. Although we can obtain this small parameter set by truncation of the coefficient set, this method does not guarantee a stable representation. In addition, it does not necessarily lead to an inverse filter whose output residual is white or one whose output residual has any other meaningful interpretation. Furthermore, when the filter parameter set is known a priori to be small (as is the case in our applications to follow) the required 2-D spectral factorization is computationally inefficient because the entire covariance function $r_x(\ell,k)$ is needed.

An alternate model described in the next section controls the inverse filter output and the coefficient mask size by guaranteeing a minimum-variance residual for a finite coefficient geometry. For low-order models, the minimum-variance representation leads to a more efficient procedure for solving the factorization problem by specifically using the finite-order model assumption.

2.3 Minimum-Variance Linear Prediction Models

Rather than viewing (3) as an input-output relation for generating $x(n,m)$, we can interpret (3) as a prediction operation. In particular, if we estimate the sample $x(n,m)$ by

$$\tilde{x}(n,m) = \sum_{\ell,k \in M} a(\ell,k)\ x(n-\ell,\ m-k) \tag{6}$$

then $w(n,m)$ in (3) is the residual process $x(n,m) - \tilde{x}(n,m)$ equal to the error in prediction. Suppose now that $x(n,m)$ is an arbitrary stationary 2-D random field. We seek to fit a linear prediction model to $x(n,m)$ that will minimize the variance of the error between the actual values of $x(n,m)$ and the predicted values $\tilde{x}(n,m)$ obtained from (6). Thus, we want to minimize

$$E[(x(n,m) - \tilde{x}(n,m))^2] \tag{7}$$

by varying the linear prediction coefficients $a(\ell,k)$. Minimization of (7) yields a set of linear equations, known as the Normal equations, to be solved for the prediction coefficients [27]. These can be expressed as

$$r_x(\ell,k) - \sum_{p,q \in M} a(p,q)\ r_x(\ell-p,\ k-q) = (\sigma_w)^2 \delta(\ell,k) \tag{8}$$

where the $r_x(\ell,k)$'s are the covariance coefficients defined in (2b). To solve (8), the geometry of the mask M in (6) associated with the prediction coefficients must be choosen. As in the WNDR, the choice of the mask M presupposes a certain relation between an image pixel and its neighbors. Although this relation need not be a causal one (see Figure 3), we shall assume (unless indicated otherwise) a NSHP mask geometry, so that each pixel is related only to neighbors in its past.

We now consider some of the properties of the minimum-variance prediction representation (MVPR). First, unlike the normal equations for 1-D causal filters, the solution to (8) does not generally result in correlation matching. That is, x(n,m) in (6) does not necessarily have the specified correlation $r_x(\ell,k)$ over any region in space [27]. Furthermore, again unlike the 1-D case, the solution to (8) does not necessarily yield a stable filter [27]. However, under certain conditions, the solution to (8) does result in important correlation (and spectral) matching properties which can most clearly be seen from a comparison of the WNDR and MVPR models.

Let us first define the error residual of linear prediction as

$$e(n,m) = x(n,m) - \tilde{x}(n,m) \tag{9}$$

$$= x(n,m) - \sum_{\ell,k \in M} a(\ell,k)x(n-\ell,m-k)$$

In general, e(n,m) is not a 2-D white noise process; that is, it has a correlation function $r_e(\ell,k)$ not equal to the 2-D impulse function. Consequently, a MVPR is generally not a WNDR. In fact, x(n,m) can be viewed as the output of a linear shift-invariant filter with a colored noise input.

However, there are some conditions under which the two representations are equivalent.

Consider first the case where x(n,m) can be characterized by a WNDR with a corresponding NSHP coefficient mask M'. If the mask M in (8) equals M', the solution to (8) yields the WNDR (3). If M' is infinite in extent then the MVPR approaches the WNDR as the mask M is allowed to become arbitrarily large [28]. Thus, the MVPR can always be made aribtarily close to the WNDR. This implies that for a sufficiently large M the spectral density of the MVPR can be made arbitrarily close to the spectral density of the WNDR. Therefore, the factorization in (5a) can be performed indirectly through (8). In the case where the mask M does not match M' (i.e., M is smaller than M') the MVPR is not WNDR. In this case, the random process x(n,m) in (9) is the output of a linear filter with a colored noise input e(n,m) and so the spectral matching property in (5a) does not hold. As a final point, we mention that the process x(n,m) may have been generated by an underlying WNDR having a noncausal coefficient mask M, as illustrated in Figure 3. However, regardless of this generation process (or any other generation process) a causal white-noise driven MVPR can always be found provided its NSHP coefficient mask is made sufficiently large [28].

As pointed out in the previous section, the MVPR provides an efficient procedure for filter coefficient determination where in our applications a low-order model is known a priori. A small model order implies a small set of linear equations. Further, depending on the mask geometry, the matrix representation of these equations has a block Toeplitz structure and

therefore are amenable to soluton by fast computational methods [29].

2.4 Model Parameter Estimation

In all of the above representations the covariance coefficients $r_x(\ell,k)$ were required. Since these parameters are not necessarily known *a priori*, we must estimate them from the available data. In general, these estimates are approximate since only a finite data segment is usually available. Even when an arbitrarily large data set is available, lack of stationarity may prohibit the use of arbitrarily long averaging.

The estimation approaches typically replace the expectation in (2b) by a sum of lagged products given by

$$r_x(\ell,k) = \sum_{p,q} x(p,q)\, x(\ell+p,k+q) \qquad (10)$$

with various limits of summation [30, 31]. One of the following two methods is most often used. In the first, it is assumed that the data is zero outside of the given set of points or data "window". In the second, the limits of summation in (10) are not allowed to run beyond the given data so that no assumptions about the data are made outside of the data window. These covariance estimates together with the Normal equations are known as the "correlation method" and the "covariance method" of linear prediction respectively [31, 32]. In the applications to follow both methods are used. The "correlation method" is used in image segmentation where large quantities of training data are available. On the other hand, the "covariance method" is used in object detection where the data is nonstationary.

2.5 Space-Variant Models

The models discussed in the foregoing sections can be made space-variant by allowing the model coefficients to be a function of position. For example, the space-variant WNDR is given by

$$x(n,m) = \sum_{(0,0)<(j,k)} \sum a(n,m;j,k)x(n-j,m-k) + w(n,m) \qquad (11a)$$

with

$$w(n,m) = \sigma_{w(n,m)}\, w_N(n,m) \qquad (11b)$$

where $w_N(n,m)$ is white Gaussian with unit variance and where both the filter coefficients and the noise variance are allowed to vary with position. The image $x(n,m)$ is now a sample function of a nonstationary random field with mean and covariance defined by (2). The various forms of this space-variant representation and corresponding estimation procedures are not as well established as in the stationary case. Nevertheless, this image representation can be very useful under a "quasi-stationary" assumption when the image background is slowly varying. This type of model is used in the object detection problem of Section 5.

3. STATISTICAL PROCEDURES

In this section, we review some basic statistical procedures which will be useful in image analysis; namely, estimation, hypothesis testing, and significance testing. It will later be seen that image segmentation requires estimation of regions and boundaries, while classification and detection requires decisions among various hypotheses. We further consider the form of the likelihood functions when a Gaussian PDF is assumed along with the linear models of the previous section.

3.1 Estimation

Assume that a set of N random variables represented by the vector $\underline{x}$ is described by a multivariate probability density function

$$p_{\underline{x};\underline{\rho}}(\underline{x};\underline{\rho}) \tag{12}$$

which is parameterized by a set of M (scalar) variables $\underline{\rho}$. The problem of parameter estimation is to estimate $\underline{\rho}$ from a given set of observations $\hat{\underline{x}}$. The two approaches referred to in this paper are maximum likelihood estimation and maximum a posteriori (MAP) estimation [33, 34].

Maximum likelihood (ML) estimation regards $\underline{\rho}$ as a deterministic parameter and chooses its value to maximize the probability that the given set of observations occurs. Specifically $\hat{\underline{\rho}}_{ml}$ is the value that satisfies

$$p_{\underline{x};\underline{\rho}}(\hat{\underline{x}};\hat{\underline{\rho}}_{ml}) = \max_{\underline{\rho}} p_{\underline{x};\underline{\rho}}(\hat{\underline{x}};\underline{\rho}) \tag{13}$$

When used in this manner the function $p_{\underline{x};\underline{\rho}}(\hat{\underline{x}};\cdot)$ of $\underline{\rho}$ is referred to as a likelihood function.

MAP estimation regards $\underline{\rho}$ as a random vector and assumes a prior density $p_{\underline{\rho}}(\underline{\rho})$. To emphasize the random nature of $\underline{\rho}$ it is conventional in MAP estimation to denote the density function (12) by $p_{\underline{x}|\underline{\rho}}(\underline{x}|\underline{\rho})$. MAP estimation chooses the estimate $\hat{\underline{\rho}}_{MAP}$ to maximize

$$\max_{\underline{\rho}} \left[p_{\underline{x}|\underline{\rho}}(\hat{\underline{x}}|\underline{\rho}) \cdot p_{\underline{\rho}}(\underline{\rho}) \right] \tag{14}$$

which is equivalent to maximizing the posterior density function

$$p_{\underline{\rho}|\underline{x}}(\underline{\rho}|\hat{\underline{x}}). \tag{15}$$

Since both ML and MAP estimation involve maximizing probabilities of events closely related to the observed data, both procedures are quite intuitively plausible. Most readers will be aware that these estimates also have important statistical properties such as consistency and unbiasedness and satisfy certain bounds on their variance or mean square error [33,34].

3.2 Hypothesis Testing

Hypothesis testing is another statistical procedure in image analysis and is useful for image classification problems where the classes have known statistical characteristics. In the case of classifying images

into one of two different categories, various criteria (Bayes, Neyman-Pearson, etc.) lead to a likelihood ratio test [33]:

$$\ell_{12} = \frac{p_1(\hat{\underline{x}}|H_1)}{p_2(\hat{\underline{x}}|H_2)} \underset{H_2}{\overset{H_1}{\gtrless}} T \tag{16}$$

where $p_i(\hat{\underline{x}}|H_i)$ is the likelihood function conditioned on H_i, the hypothesis that the observed data $\hat{\underline{x}}$ belongs to class i. The notation in (16) is used to mean that the decision is H_1 when the likelihood ratio is greater than the threshold T and the decision is H_2 when the likelihood ratio is less than the threshold T.

Generalizations of hypothesis testing to more than two classses are somewhat more difficult to derive but are nevertheless easy to apply. The simplest such test involves choosing the class for which the likelihood function $p_i(\hat{\underline{x}}|H_i)$ is largest. More generally, they involve decisions using pairs of likelihood ratios between the various classes [33].

3.3 Significance Testing

Significance testing is probably the least well known of the statistical procedures in the context of image and signal processing. Nevertheless, it is a powerful concept and is used in developing one of the algorithms discussed in this paper. In significance testing a single hypothesis is checked against all possible alternatives [34]. Thus, the procedure is important when we have good statistical knowledge about one class of images and little or no knowledge about the other classes.

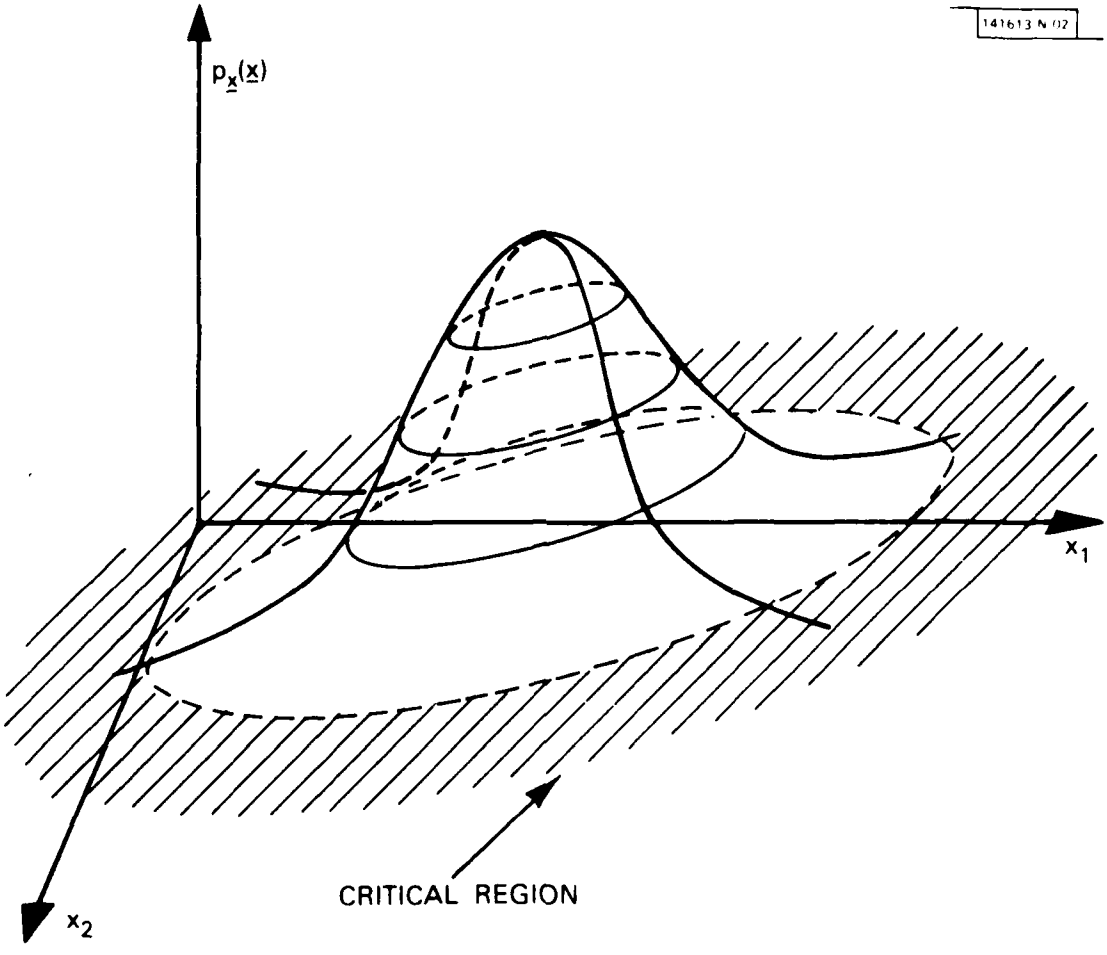


Figure 4. Illustration of significance testing.

The idea can be illustrated with a two-dimensional observation vector $\hat{\underline{x}}$. Figure 4 shows the probability density function for $\underline{x}$ based on an assumed hypothesis H_0. A so-called "critical region" of small probability is chosen arbitrarily. The probability of the event that $\hat{\underline{x}}$ falls within this region is known as the significance of the test. If the observation falls within the critical region, we reject H_0. Otherwise we accept H_0. Although the choice of the critical region is somewhat arbitrary, one may be guided by any weak statistical knowledge about the nature of the other classes.

3.4 Form of the PDF Based on Linear Models and Gaussian White Noise

The statistical procedures derived in the previous sections require the use of the data vector $\underline{x}$ with a specified PDF. Although the observed data can be used directly within these tests, such use is cumbersome and computationally inefficient for image processing applications. Furthermore, it is unsatisfying since it provides for no convenient intuitive interpretation beyond that of the statistical test itself. Alternately, when the data x(n,m) is characterized by a Gaussian PDF and follows the linear models of Section 2, it is possible to transform each test to one involving a set of independent Gaussian random variables. Under these conditions, the transformed data set is the output of a filter which is the inverse of the linear filter used to generate the random process $\underline{x}$. This mapping is the key to the statistical approaches taken within this paper and leads to tests which are insightful and simple to implement.

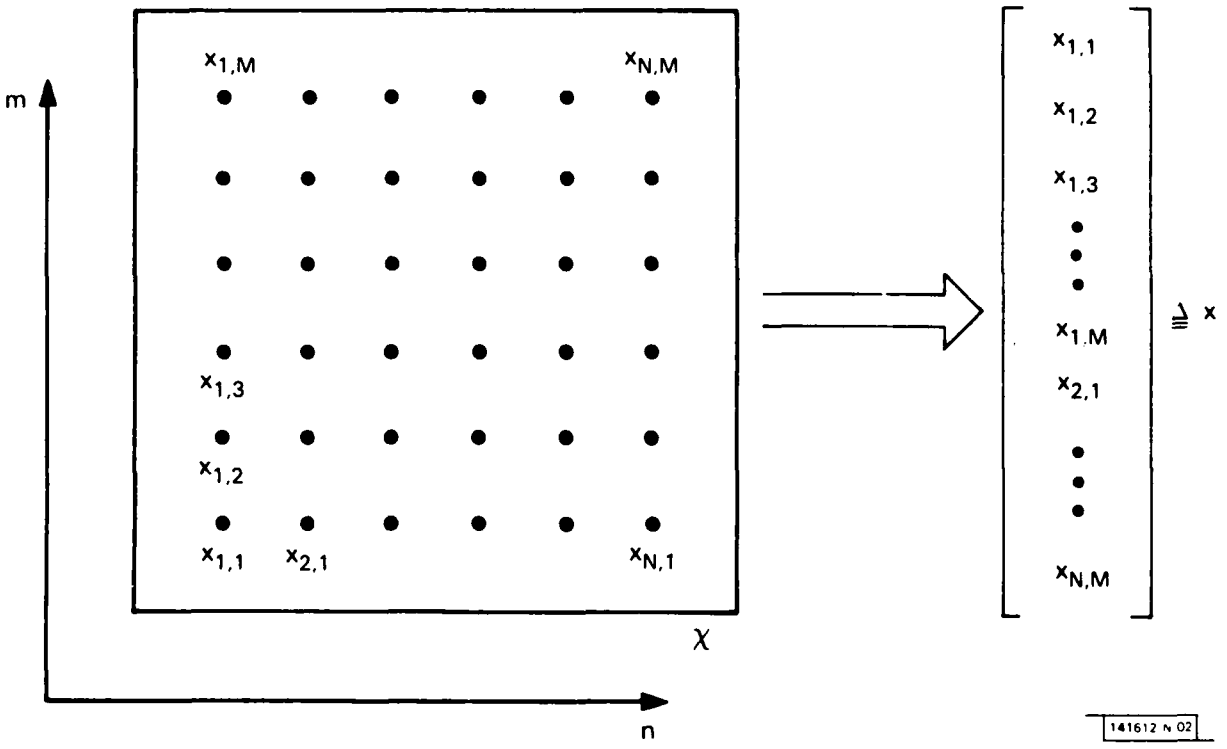


Figure 5. Formation of the data vector $\underline{x}$.

Let the NM-dimensional data vector $\underline{x}$ be formed by concatenating the columns (or rows) of a region χ of finite size N by M, as illustrated in Figure 5. Note that in Fig. 5 and the following paragraphs we use two-dimensional subscript notation to indicate position of points with respect to the region χ. In other words, if χ is located so that its lower left corner is at (n_o, m_o) then $x_{1,1}$ represents the value $x(n_o,m_o)$. Further, suppose that the elements of χ follow an autoregressive WNDR given by (3) with a finite NSHP mask and are characterized by a Gaussian PDF

$$p_{\underline{x}}(\underline{x}) = \frac{1}{(2\pi)^{NM/2}\left|K_x\right|^{1/2}} \exp\left[-\underline{x}^T K_x^{-1} \underline{x}\right] \tag{17}$$

where $K_x=E[\underline{x}^T\underline{x}]$ and where (without loss of generality) we have assumed a mean level of zero. We shall take two approaches to the problem of transforming $\underline{x}$ to a set of uncorrelated Gaussian random variables: the first is approximate, but insightful and practical, while the second approach is exact. The difference is in how we handle the boundary conditions at the border of the observed data.

If the vector $\underline{w}$ represents the white noise input term in (3) over the region χ then (3) can be written in matrix form as

$$A\underline{x} = \underline{w} - A_o\underline{x}_o \tag{18}$$

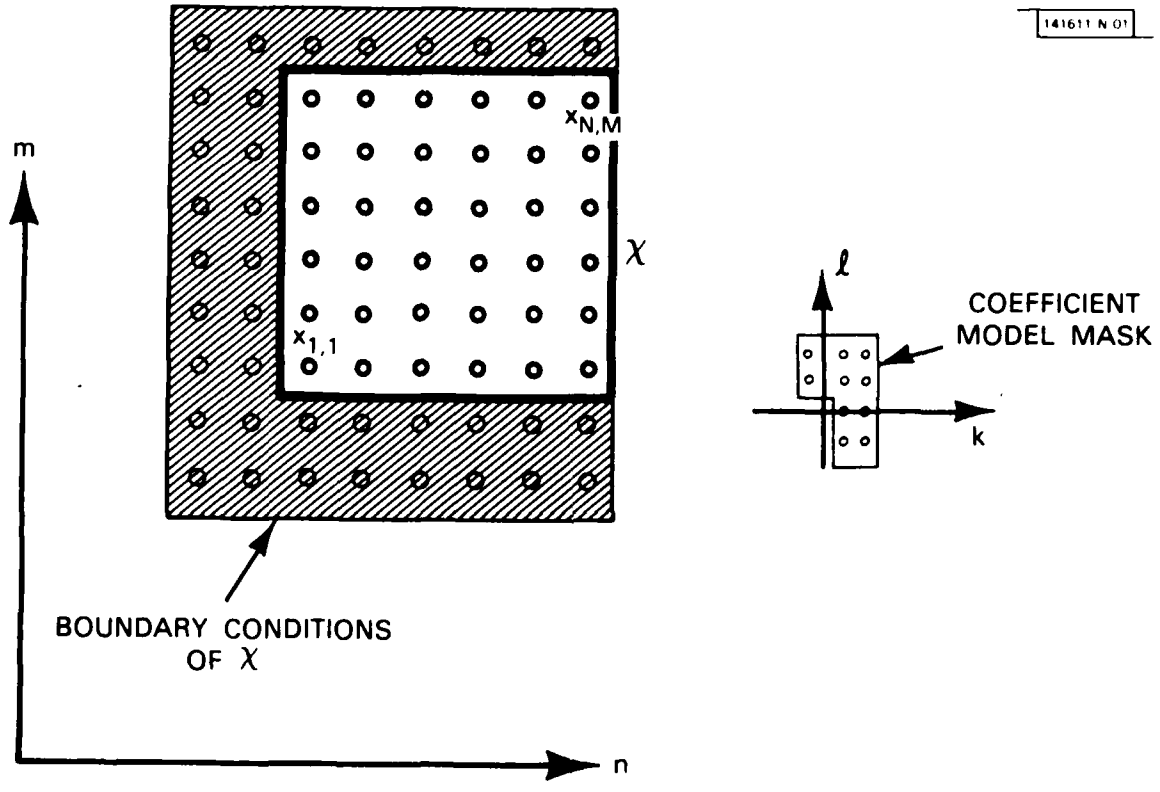


Figure 6. Illustration of boundary conditions outside the region χ.

where A and A_o are matrices whose nonzero elements are derived from the terms $a(\ell,k)$ in (3) and $\underline{x}_o$ represents a set of boundary conditions with support outside of χ illustrated in Figure 6 for a particular coefficient mask. Observe from (3) that the matrix A is square and lower triangular with ones along its diagonal. Thus, A is always nonsingular. The matrix A_o in general, however, is not square and has no special properties. If the boundary conditions are temporarily ignored, i.e., we assume $\underline{x}_o=\underline{0}$, then we have the approximate relation

$$\underline{w} \simeq A\,\underline{x} \tag{19}$$

Therefore, we can write the PDF of $\underline{x}$ as

$$p_{\underline{x}}(\underline{x}) \simeq \left|A\right|^{-1} p_{\underline{w}}(\underline{w})\Big|_{\underline{w}\,=\,A\underline{x}} \tag{20}$$

Because of the properties discussed above, the determinant of A is always equal to one. Then, since $\underline{w}$ is characterized by a Gaussian PDF, we have

$$p_{\underline{x}}(\underline{x}) \simeq \frac{1}{(2\pi)^{NM/2}\left|K_w\right|^{1/2}} \exp\left[-\frac{1}{2}\,\underline{w}^T K_w^{-1}\,\underline{w}\right]\Big|_{\underline{w}\,=\,A\,\underline{x}}$$

$$= \frac{1}{(2\pi)^{NM/2}(\sigma_w)^{NM}} \exp\left[-\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\frac{(w_{n,m})^2}{(\sigma_w)^2}\right]\Big|_{\underline{w}\,=\,A\,\underline{x}} \tag{21}$$

where $w_{n,m}$ is an element of vector $\underline{w}$ and where K_w is a diagonal matrix with diagonal elements equal to $(\sigma_w)^2$. Consequently, the likelihood functions of the previous sections can be implemented approximately with the

white noise vector $\underline{w}=A\underline{x}$. In practice, the elements of $\underline{w}$ can be obtained by inverse filtering $x(n,m)$ by the filter with transfer function $A(z_1, z_2)$.

The reader should bear in mind that because of (19), relation (21) is only approximate. An alternate argument that yields an exact result relies on removing any assumptions about the boundary conditions outside of the image support χ. To achieve the exact result we begin by performing a triangular decomposition of the covariance matrix K_x. In particular, since the covariance matrix K_x is symmetric and positive semi-definite, K_x can be uniquely factored in terms of upper triangular, lower triangular, and diagonal matrices [35]:

$$K_x=LDL^T \tag{22}$$

In this equation L is a lower triangular matrix with one's along its diagonal and D is a diagonal matrix. Solving (22) for D yields

$$L^{-1}K_x(L^{-1})^T = D \tag{23}$$

and thus D is recognized as the covariance matrix of a set of Gaussian uncorrelated random variables $\underline{v}$ related to $\underline{x}$ by the linear transformation

$$\underline{v}=L^{-1}\underline{x} \tag{24}$$

It is straightforward to show that since L is lower triangular with unit diagonal, L^{-1} has the same property and thus (24) represents a causal transformation of the vector $\underline{x}$ in the following sense. Suppose we consider a subset $x_{1,1}, x_{1,2}, \ldots, x_{n,m-1}$ of the points in χ as shown in Fig. 7.

These points represent the points in the vector $\underline{x}$ appearing before $x_{n,m}$ (see Fig. 7). Then since L^{-1} is lower triangular each vector component $v_{n,m}$ is a function of only these points and $x_{n,m}$.

Now proceeding as in the previous derivation, we can write the PDF of $\underline{x}$ as

$$p_{\underline{x}}(\underline{x}) = |L| p_{\underline{v}}(\underline{v}) \Big|_{\underline{v} = L^{-1}\underline{x}} \tag{25a}$$

and since $|L| = 1$, we have

$$p_{\underline{x}}(\underline{x}) = \frac{1}{(2\pi)^{NM/2} |D|^{1/2}} \exp\left[-\frac{1}{2}\, \underline{v}^T D^{-1}\, \underline{v}\right] \Big|_{\underline{v}=L^{-1}\underline{x}} . \tag{25b}$$

The factorization in (22) leading to the transformation in (24) is of interest since it can be shown that the rows of L^{-1} are coefficients of the causal minimum variance prediction solution of orders 1 through NM and that the diagonal elements of D are the corresponding prediction error variances [35]. Figure 7 illustrates how the prediction errors $v_{n,m}$ are generated as a causal function of the points of χ. Thus, each row of L^{-1} represents the coefficient set of a minimum-variance model with NSHP mask support.

Finally, then, returning to (25b) we have

$$p_{\underline{x}}(\underline{x}) = \frac{1}{(2\pi)^{NM/2} \prod_{n=1}^{N} \prod_{m=1}^{M} \sigma_{v_{n,m}}} \exp\left[-\frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{M} \frac{(v_{n,m})^2}{(\sigma_{v_{n,m}})^2}\right] \Bigg|_{\underline{v}=L^{-1}\underline{x}} . \tag{26}$$

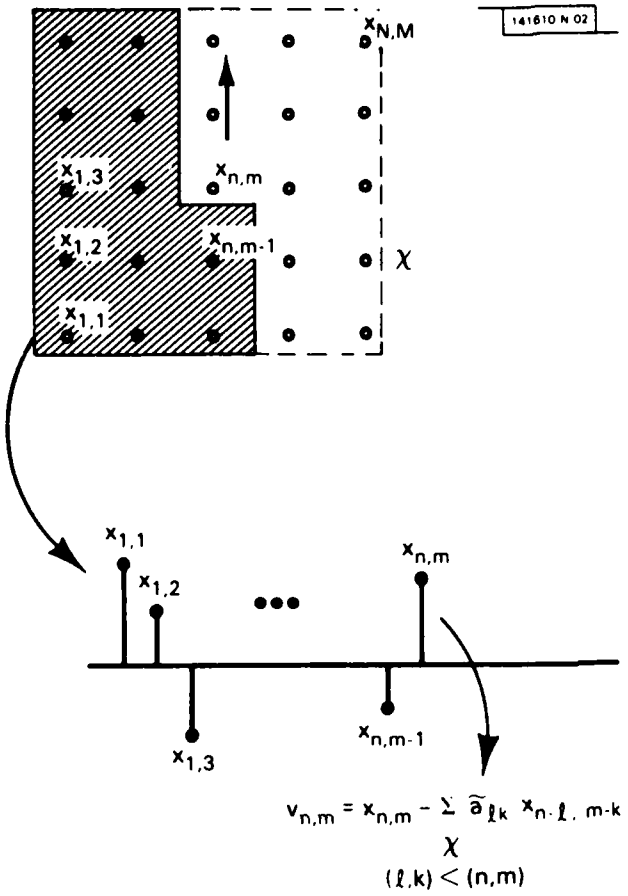


Figure 7. Growing prediction of $x_{n,m}$ from its previous values.

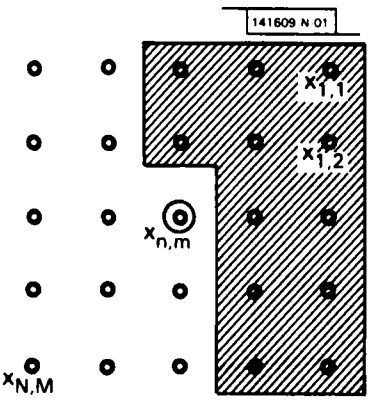


Figure 8. "Anti-casual" prediction of $x_{n,m}$.

The elements $v_{n,m}$ of $\underline{v}$ can be interpreted as prediction errors and the $(\sigma_{v_{n,m}})^2$ are the corresponding prediction error variances which generally will differ as the prediction order changes.

Let us now compare the results (21) and (26). The transformations given by (19) and (24) and the corresponding PDF's (21) and (26) have the same form but (24) and (26) represent an exact whitening process. The vector $\underline{w}$ is obtained from a fixed inverse (prediction) filter corresponding to the WNDR, while $\underline{v}$ is obtained from a growing predictor corresponding to the minimum-variance representation of increasing order. However, as the order of the MVPR increases so that its mask support equals that of the WNDR then the whitened elements of the two transformations and their respective variances become the same. When the image support χ is much larger than the order of the WNDR this equivalence holds almost everywhere over χ except near its border. For this reason and since computation of growing predictors is often intractable in practice, use of the vector $\underline{w}$ is found to be preferred.

It is curious to note that although (22) was derived with no imposed directionality, (24) implies a causal relation among data elements. This apparent contradiction can be resolved by noting that the causality of the growing predictor in (24) arises because of our way of ordering samples $x(n,m)$ in support χ. For example, suppose we order samples in reverse order as illustrated in Figure 8. With this ordering, the predictors become "anti-causal". As before, we can approximate these growing predictors with fixed-order predictors.

4.0 APPLICATIONS - CLASSIFICATION AND SEGMENTATION

In this section we consider the two related problems of image classification and segmentation. We assume throughout this section that the images to be considered contain one or more regions of homogeneous texture and are well modeled by the linear models described in the Section 2. This was found to be the case in dealing with aerial photographs of natural terrain. In images of natural terrain such regions may represent various types of fields, water, desert, wooded areas, and so on. The problem of image classification is as follows. Given an image containing only one of these region types, determine the class, among several known classes, to which the image belongs. The problem of image segmentation is similar, but more complex. Given an image consisting of several different homogeneous regions, determine the extents of these regions and their classes. We shall approach these problems by using the linear filtering models to characterize the images and by applying the statistical methods of Section 3 to develop appropriate algorithms. Throughout, we will assume that the regions are large enough that we can ignore the effects of the boundary conditions.

4.1 Image Classification

Assume that it is desired to classify images of a fixed size N by M into one of two distinct classes. Each class is modeled by an all-pole filter driven by Gaussian white noise, and all of the model parameters are known. Then according to Section 3, an optimal procedure for classifying the images is to use the likelihood ratio test (16) where $\underline{\hat{x}}$ represents a vector of the image data to be classified. Since each class of

images is assumed to have an all-pole linear filtering model, (21) applies.

Let $w^i_{n,m}$ be the prediction error residual from linear predictive (inverse) filtering with the filter of class i and let $(\sigma_1)^2$ be the corresponding prediction error variance. Then from (16) and (21) the likelihood ratio test is

$$\ell_{10} = \frac{\frac{1}{(2\pi)^{NM/2}(\sigma_1)^{NM}} \exp\left[-\sum_{\chi} \frac{(w^1_{n,m})^2}{2(\sigma_1)^2}\right]}{\frac{1}{(2\pi)^{NM/2}(\sigma_0)^{NM}} \exp\left[-\sum_{\chi} \frac{(w^0_{n,m})^2}{2(\sigma_0)^2}\right]} \underset{H_0}{\overset{H_1}{\gtrless}} T \qquad (27)$$

where we have dropped the subscript w on σ for notational clarity and added appropriate class subscripts.

Taking minus twice the logarithm leads to the test

$$\sum_{\chi} \frac{(w^1_{n,m})^2}{(\sigma_1)^2} - \sum_{\chi} \frac{(w^0_{n,m})^2}{(\sigma_0)^2} + NM \ln \frac{(\sigma_1)^2}{(\sigma_0)^2} \underset{H_0}{\overset{H_1}{\lessgtr}} -2\ln T \qquad (28)$$

Eq. (28) states that if the sum of the normalized squared errors in linear prediction is lowest for class i, then that class should be selected. This interpretation is intuitively satisfying. Further, it can be shown that regardless of the statistical distribution of the residuals, a decision rule in the form of (28) will, on the average, choose the correct class [2].

When ther are more than two classes of images to deal with, the decision rule becomes only slightly more complicated. As stated in Section 3.2, the multiple class problem involves likelihood ratio tests between pairs of classes. Each of these likelihood ratio tests thus has the general form of (28).

4.2 Image Segmentation

We now turn to the problem of image segmentation. While the statistical basis for image classification was hypothesis testing, the statistical basis for segmentation will be estimation theory. We find that the quantities to be estimated are a set of class labels or "states" for each of the pixels. These states collectively determine the regions or segments which we seek to find.

We shall consider both ML and MAP estimates for the states. Recall that in order to obtain a MAP estimate of the states, we need to have some prior density function. It is found that this density function is most conveniently obtained if the states themselves satisfy a Markov probability model. This Markov model forms a superstructure for the image built over the individual linear filtering models that describe the image within each region. The form of the Markov model is discussed in the next section.

4.2.1 Model Development

Consider an image consisting of multiple regions R_1, R_2,...,R_q (see Fig. 9). It will be assumed that within each region, the image can be represented by a white-noise driven autoregressive model as in (3). Let the

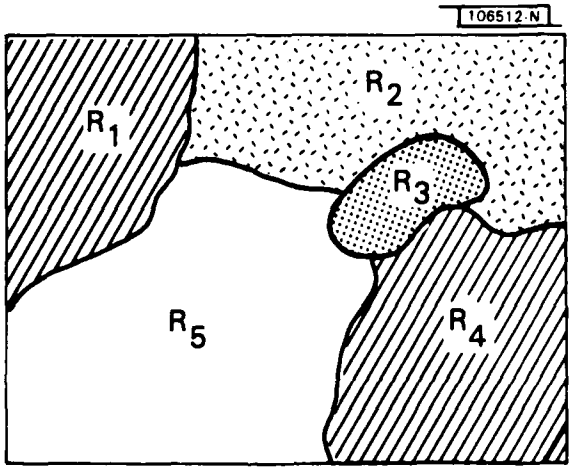


Figure 9. Image consisting of multiple textured regions.

image in region R_i be represented by a model of class k_i and denote the probability density function for points in that region by $p_{k_i}(\underline{x}_{R_i})$ where $\underline{x}_{R_i}$ is the set of points in region R_i. Since points falling in disjoint regions are generated by different linear filter models and thus are independent, we can write the probability for all points $\underline{x}$ in the image, given the regions R_1, R_2, ..., R_q as

$$p(\underline{x}|R_1,..,R_q) = p_{k_1}(\underline{x}_{R_1}) \cdot p_{k_2}(\underline{x}_{R_2}) \cdots \cdot p_{k_q}(\underline{x}_{R_q}) \tag{29}$$

If we use the relation (21), we can obtain an explicit expression for the density function in terms of error residuals of linear prediction for each of the image models. Let us follow the procedure in Section 4.1 and instead form the log likelihood function $-2 \ln p(\hat{\underline{x}}|R_1 \ldots R_q)$. Then from (29) and (21) we have

$$-2 \ln p(\underline{x}|R_1,R_2,\ldots,R_q)$$

$$\approx \sum_{R_1} \left[\frac{(w^{k_1}_{n,m})^2}{(\sigma_{k_1})^2} + \ln(\sigma_{k_1})^2\right] + \ldots + \sum_{R_q} \left[\frac{(w^{k_q}_{n,m})^2}{(\sigma_{k_q})^2} + \ln(\sigma_{k_q})^2\right] - NM \ln 2\pi$$

$$= \sum_{i=1}^{q} \sum_{R_i} \left[\frac{(w^{k_i}_{n,m})^2}{(\sigma_{k_i})^2} + \ln(\sigma_{k_i})^2\right] - NM \ln 2\pi$$

where $\hat{\underline{x}}$ is the observed data vector, and where $w^{k_i}_{n,m}$ is the error residual computed using a filter of class k_i

and $(\sigma_{k_i})^2$ is the white noise variance of class k_i.

Eqs. (29) and (30) suggest that one can take a maximum likelihood approach to estimating the regions. For ML estimation, the number of regions q and the regions themselves are considered to be deterministic parameters of the density function. An ML estimate for these parameters is obtained by choosing values that maximize (29) or, equivalently, minimize (30). It is clear from (30) that the function is minimized if every point (n,m) in the image is assigned to a region R_i of type k_i such that the term in brackets is minimum. Suppose there are two possible region types so that k_i takes on values 0 and 1. Then we are led to a segmentation rule of the form

$$h_0 \underset{1}{\overset{0}{<>}} h_1 \tag{31a}$$

with

$$h_k = \frac{(w^k_{n,m})^2}{(\sigma_k)^2} + \ln(\sigma_k)^2 \quad k = 0,1 \tag{31b}$$

where the number above or below the inequality indicates the region type to which the point (n,m) will be assigned if the inequality holds.

Since ML estimation leads to a decision rule that assigns points to region types without regard to the assignment of adjacent points, one might expect this algorithm to produce a number of false assignments and a somewhat "spotty" result. This, in fact is seen to be the case.

A better procedure is to use MAP estimation as discussed in Section 3.2. For MAP estimation the regions are considered to be random quantities, and we maximize the probabilty for a given set of regions conditioned on our observation of the image. From Bayes rule, the a posteriori probability can be written as

$$\Pr\left[R_1, R_2 \ldots, R_q \middle| \hat{\underline{x}}\right] = \frac{p(\hat{\underline{x}} \middle| R_1, R_2, \ldots, R_q) \Pr\left[R_1, R_2 \ldots, R_q\right]}{p(\hat{\underline{x}})} \tag{32}$$

Maximizing (32) requires maximizing the numerator which in turn requires that we have an expression for the prior probability of the regions. Thus we are led to the following procedure for modeling the region statistics.

Let us define the "state" s(n,m) of a point (n,m) as the region type to which that point has been assigned. If there are K region types, the state will take on values in the set of integers 0,1,2,...K-1. For the subsequent development, we shall consider the case of K = 2. Generalizations are possible, of course. Since the set of all possible state assignments for points in the image is one-to-one with the set of all possible divisions of the images into regions, the region estimation problem can be viewed as one of estimating the states of the points. Now, assume the state of a point is stochastically dependent on some adjacent set of states $S_{n,m}$ in a symmetric

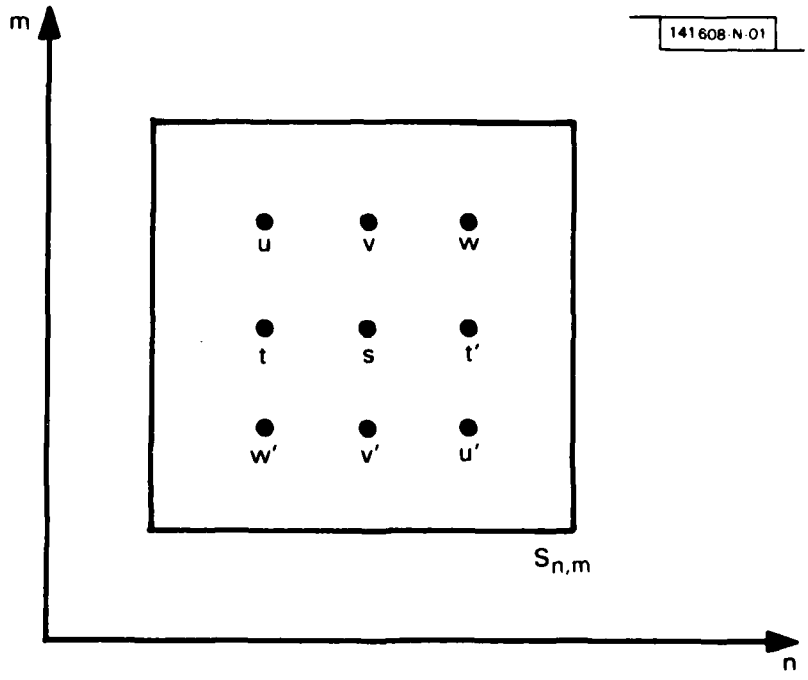


Figure 10. State dependence of the point s on its adjacent symmetric neighbors.

support region as shown in Fig. 10. Let $S_{\underline{x}}$ represent a chosen set of state assignments for all points in the image. Let us denote by $Pr[S_{\underline{x}}]$ the joint probability that the points in the image take on the chosen set of state assignments $S_{\underline{x}}$. We would like to find a probability structure on the states so that the probability of $S_{\underline{x}}$ can be written as

$$Pr[S_{\underline{x}}] = \prod_{(n,m)} Pr[s(n,m)|S_{n,m}] \tag{33}$$

where the terms in the product are intended to represent the probability that the state of point (n,m) takes on a particular value s(n,m) given that the surrounding points assume the set of values $S_{n,m} \subset S_{\underline{x}}$. There is a very general class of Markov processes that allows the representation (33) and whose properties have been studied in detail [36]. A particular form that is allowed is

$$Pr[s(n,m)|S_{n,m}]\exp[s(n,m)\{\alpha+\beta_1(t+t')+\beta_2(v+v')+\gamma_1(u+u')+\gamma_2(w+w')\}] \tag{34}$$

where t=s(n-1,m), t' = s(n+1,m) and the other variables are similarly defined as shown in Fig. 10. The parameters $\alpha, \beta_1, \beta_2, \gamma_1$, and γ_2, are arbitrary constants and D is a normalizing constant. More general forms of the probability are allowed [see Ref. 36], but (34) will be sufficient for our purposes here. The terms $Pr[s(n,m)|S_{n,m})]$ are interpreted as Markov transition probabilities. The constants $\alpha, \beta_1, \beta_2, \gamma_1, \gamma_2$, provide a fair amount of flexibility since the transition probabilities can be made dependent on the horizontal, vertical, and diagonal

directional patterns of the surrounding states to any desired degree. One particular selection of the parameters, however, namely $\alpha = -4$, $\beta_1=\beta_2=\gamma_1=\gamma_2=1$, which gives equal weight to all directions, leads to a particularly simple and interesting algorithm. In this case, we have

$$\Pr\left[1 \middle| S_{n,m}\right] = \frac{1}{D} \exp \sum_{(i,j)\in S_{n,m}} (s(i,j) - 1/2) \tag{35a}$$

$$\Pr\left[0 \middle| S_{n,m}\right] = \frac{1}{D} \tag{35b}$$

Returning to (32) let us observe that since the set of all possible region assignments for an image is one-to-one with the set off all possible state assignments for the pixels, the second term in the numerator of (32) can be replaced by $\Pr[S_{\underline{x}}]$. Thus maximizing (32) is equivalent to minimizing

$$2 \ln p(\underline{x} | R_1 \ldots R_q) - 2 \ln \Pr[S_{\underline{x}}] \tag{36}$$

Further, observe that since $s(n,m) = k_i$ for $(n,m) \in R_i$ then by simply reordering terms in the sums, the double summation in (30) can be written as

$$\sum_{i=1}^{q} \sum_{R_i} \frac{(w_{n,m}^{k_i})^2}{(\sigma_{k_i})^2} + \ln (\sigma_{k_i})^2 = \sum_{n=1}^{N} \sum_{m=1}^{M} \frac{(w_{n,m}^{s(n,m)})^2}{(\sigma_{s(n,m)})^2} + \ln(\sigma_{s(n,m)})^2 \tag{37}$$

By substituting (33) and (37) into (36) we find that the MAP estimate requires that the pixel states be chosen to minimize

$$\sum_{n=1}^{N}\sum_{m=1}^{M}\frac{\left(w_{n,m}^{s(n,m)}\right)^2}{\left(\sigma_{s(n,m)}\right)^2}+\ln\left(\sigma_{s(n,m)}\right)^2-2\ln\Pr\left[s(n,m)\middle|S_{n,m}\right]. \tag{38}$$

The minimization of (38) is a very difficult problem, particularly because of the Markov dependency of a state on all of its immediate neighbors. However, if we again assume that there are only two possible region types then a possibly suboptimal solution to (38) can be obtained by requiring the conditions

$$\frac{\left(w_{n,m}^{0}\right)^2}{\left(\sigma_0\right)^2}+\ln\left(\sigma_0\right)^2-2\ln\Pr\left[0\middle|S_{n,m}\right]\underset{1}{\overset{0}{\lessgtr}}\frac{\left(w_{n,m}^{1}\right)^2}{\left(\sigma_1\right)^2}+\ln\left(\sigma_1\right)^2-2\ln\Pr\left[1\middle|S_{n,m}\right] \tag{39}$$

to be satisfied simultaneously for all (n,m)[1]. Solution of (39) by direct methods for any reasonably sized images appears to be hopeless since all of the NM states are coupled through nonlinear equations. However, we can attempt to satisfy the inequalities by iteration as follows. An initial set of states, the maximum likelihood states, are computed from (31) and assigned

[1]If the Markov transition probabilities are defined by (35), the computing the terms $-2\ln\Pr\left[i\middle|S_{n,m}\right]$ is equivalent to counting the number of states in $S_{n,m}$ that have value i and dividing by the total number of states in $S_{n,m}$.

to the image points. The Markov transition probabilities are then evaluated for these state assignments and held fixed, while the inequalities are evaluated to determine a new set of states. These new state estimates are then inserted into the inequalities, and the procedure is repeated in an iterative manner until the state assignments no longer change. Although we have not yet been able to state conditions for which the iteration will converge, convergent solutions have been obtained in most cases after about ten iterations. This solution procedure bears some resemblance to various relaxation labeling techniques [37,38] in that the state assignments are iteratively updated by considering the state assignments of neighboring pixels.

4.2.2 An Example

Figure 11 shows a digitized aerial photograph of a rural area containing trees and fields. The digitized image size is 128 x 128 pixels, with gray levels represented on a scale of 0 to 255 (eight bits). Filters for each terrain class were designed using training data and applied to the image to perform the segmentation [2]. Although a nonsymmetric half-plane filter should be required, in general, our experiments showed that comparable results could be obtained by using a 4 x 4 pixel quarter-plane filter.

The filter parameters are estimated by computing the covariance matrix from a set of data and solving the Normal equations. In this case, the "correlation method" of linear prediction was used to compute the covariance matrix.

Figure 11 shows the results of segmenting the image into four categories: two types of fields, large trees and small trees. The segmentation was effected by first performing the two-class segmentation into field and tree regions and then performing additional segmentation within each region. Region boundaries, overlaid in Fig. 11, are observed to be quite accurate.

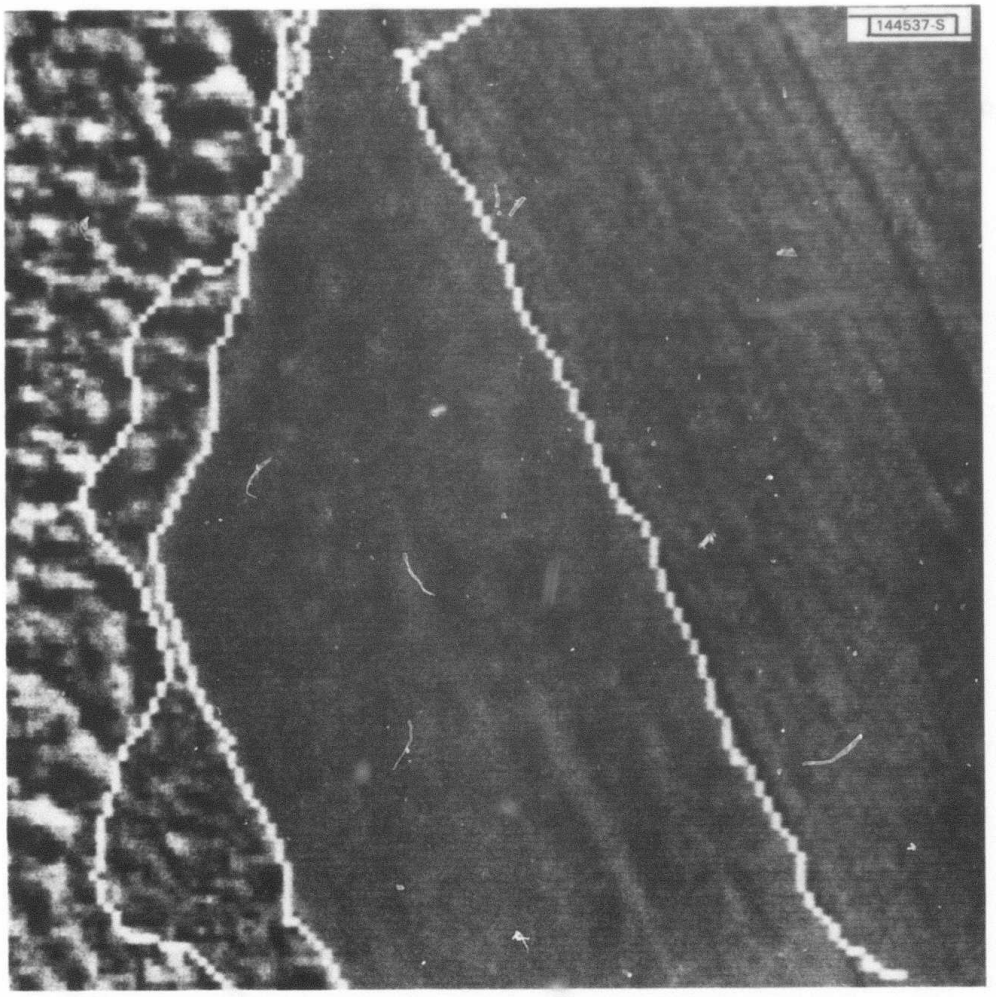


Figure 11. Segmentation of field and tree image into four categories illustrated with overlayed boundaries.

5. APPLICATIONS - OBJECT DETECTION

The problem of detecting small areas of images which differ in some statistical sense from their immediate surroundings is of considerable interest in a number of image processing applications. This section addresses the problem of detecting anomalous areas (such as objects) in backgrounds of grass, fields, or trees in aerial photographs. It was argued in the previous section that a WNDR provides a useful framework for image analysis in such stationary textured regions. In order to establish a formal procedure for detection, we generalize this representation by viewing an image background as a sample function of a 2-D nonstationary random field. This model accounts for the possibly space-varying characteristics of the textured backgrounds. It is assumed that the statistics of object pixels within this 2-D field are unknown (since it is desired to detect a broad class of objects), but that the background statistics are known or can be estimated. Our decision process relies on the significance testing procedure of Section 3.3 and is designed to ensure constant false alarm rate (CFAR) detection although background statistics may generally be changing.

5.1 Significance Testing for Object Detection

Let us suppose that an image background process is represented by a WNDR in which anomalous areas are imbedded. Further, let χ denote a small sliding region in the image. The elements of χ are concatenated by columns (or rows) into the observed data vector $\hat{\underline{x}}$. We want to decide whether the samples in χ correspond entirely to the modeled random process with probability density function $p_{\underline{x}}(\underline{x})$ or whether χ contains something other than the background random field. We shall refer to χ as the "decision region" and

our decision will be associated with the center[2] pixel location (n_o, m_o) of χ.

To make this decision, we shall apply a significance test for which a critical region R in the space of possible values for $\underline{x}$ is defined by the condition

$$p_{\underline{x}}(\underline{x}) < \lambda \tag{40}$$

In this significance test, if (40) is satisfied, i.e., if the data vector $\hat{\underline{x}}$ falls within the critical region, we decide an object is present within χ.

Suppose that after removal of the mean that the background image process is nonstationary and is given by the space-variant WNDR of (11). In this case, the covariance matrix K_x of the background has no special structure, but we shall assume it is known or can be estimated. Further, suppose that the background corresponds to the Gaussian random process in (17). Then by taking the logarithm of (40) and simplifying, the significance test becomes:

$$\text{"If } \hat{\underline{x}}^T K_x \hat{\underline{x}} > f(K_x, \lambda) \text{ , then decide object present"} \tag{41a}$$

with the threshold given by

$$f(K_x, \lambda) = \ln\left[(2\pi)^{NM}\left|K_x\right|\right] - 2 \ln \lambda \tag{41b}$$

---

[2]We assume χ is of size MxM pixels where M is odd.

Note that the size of the matrix K_x is proportional to the size of the region χ. Because the image background is nonstationary, the matrix K_x must be estimated at each pixel.

Now consider transforming the significance test (41) to one involving a set of independent random variables. Observe that the results of Section 3.4 were not limited to stationary data, i.e., the covariance matrix K_x need not have any special structure to arrive at the modified PDF's (21) and (26). Then with the exact whitening transformation leading to (26), we can write the significance test (40) as [3,4]

$$\text{"If } \sum_{\chi} \frac{(\hat{v}_{n,m})^2}{(\sigma_{v_{n,m}})^2} > f(D,\lambda), \text{ then decide object present"} \qquad (42)$$

where the function $f(\cdot,\cdot)$ is defined in (41b), where D and $\sigma_{v_{n,m}}$ are defined in Section 3.4, and where the elements $\hat{v}_{n,m}$ denote the linear prediction residuals computed from the observation vector $\underline{\hat{x}}$. Recall that the linear predictors are of increasing order (see (24)) and are derived from a minimum-variance prediction representation; the $\sigma_{v_{n,m}}$'s are the corresponding prediction error variances. Since the background process is nonstationary, each predictor and its residual variance is now a function of the location of the region χ, as well as the predictor order.

5.2 Constant False Alarm Rate Detection

Since the image background is assumed to be nonstationary, the covariance computed for the sliding region χ changes as a function of position in the image. This change in the covariance function corresponds to a change in the

PDF (17), and thus to a non-uniform probability of false alarm. We formally define the probability of false alarm, P_F, by

$$P_F = \int_R p_x(x)\, dx \tag{43}$$

where R is a critical region in the observation space specified by (40). In order to maintain a constant false alarm rate (CFAR), we must vary λ in accordance with changes in the covariance matrix K_x of (17) so that the integral in (43) remains constant.

To derive the functional form of λ required for CFAR detection, we rely on the whitening transformation (24). As in the previous sections, we assume the mean has been removed. Since the residual vector in (24), derived from the background process, consists of a set of uncorrelated Gaussian random variables with covariance matrix D, we can form a vector $\underline{e}$ with the same properties, but with unit variance:

$$\underline{e} = D^{-1/2}\, L^{-1}\, \underline{x} \tag{44}$$

Then, with the substitution of (44) into (43), it is possible to express the CFAR condition as [4]

$$\begin{aligned} P_F &= \int_R P_x(\underline{x})\, dx \\ &= \int_{R'} \frac{1}{(2\pi)^{NM/2}} \exp\left[-\frac{1}{2}\, \underline{e}^T\, \underline{e} \right] d\underline{e} \end{aligned} \tag{45}$$

where P_F is constant and where R' is the transformed critical region given by $p_{\underline{x}}(LD^{1/2}\underline{e}) < \lambda$. Figure 12 illustrates the transformation from $\underline{x}$ to $\underline{e}$ for the two-dimensional case.

The integrand in (45) represents the PDF of the transformed vector $\underline{e}$ and does not depend on the covariance of $\underline{x}$. The boundary of integration, however, does depend on K_x through L and D, and is given by

$$p_{\underline{x}}(LD^{1/2}\underline{e}) = \lambda \tag{46}$$

which, from (17), can be expressed as (see Figure 12)

$$-\frac{1}{2}\ \underline{e}^T\ \underline{e} = \ln\left[(2\pi)^{NM/2}\left|D\right|^{1/2}\lambda\right] \tag{47}$$

If we can adjust λ so that the boundary doesn't vary as a function of L and D, then the probability of false alarm will be constant. Thus, to maintain a constant false alarm rate, we must have:

$$\lambda = \frac{\text{constant}}{(2\pi)^{NM/2}\left|D\right|^{1/2}} \tag{48}$$

Finally, substituting (48) into (42), we obtain the CFAR detector

"If $\sum_{\chi} \frac{(\hat{v}_{n,m})^2}{(\sigma_{v_{n,m}})^2} > -\ln\left[(2\pi)^{NM}\left|D\right|\right] + \ln\left[(2\pi)^{NM}\left|D\right|\right] + \text{constant}$,

then decide object present" (49)

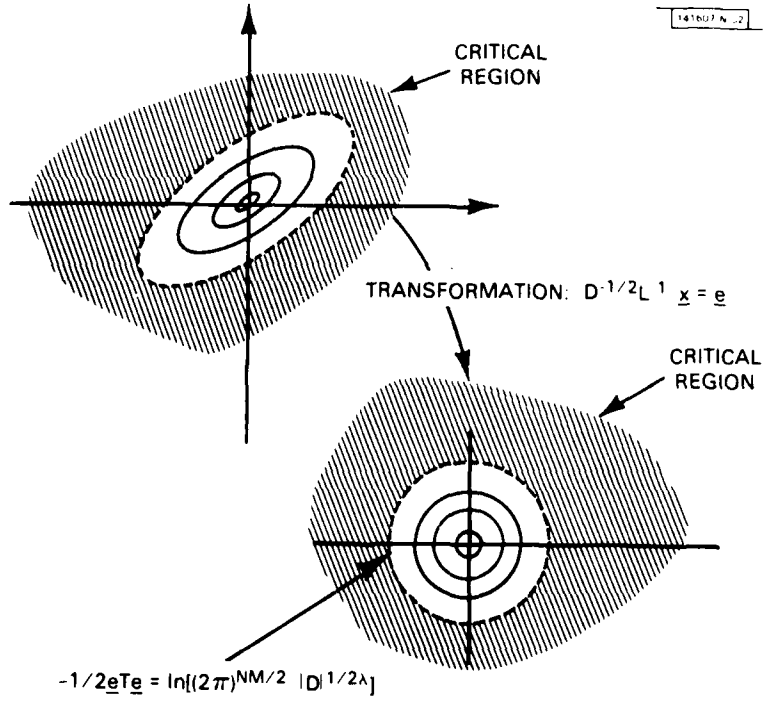


Figure 12. Pictorial representation of transformation of the critical region.

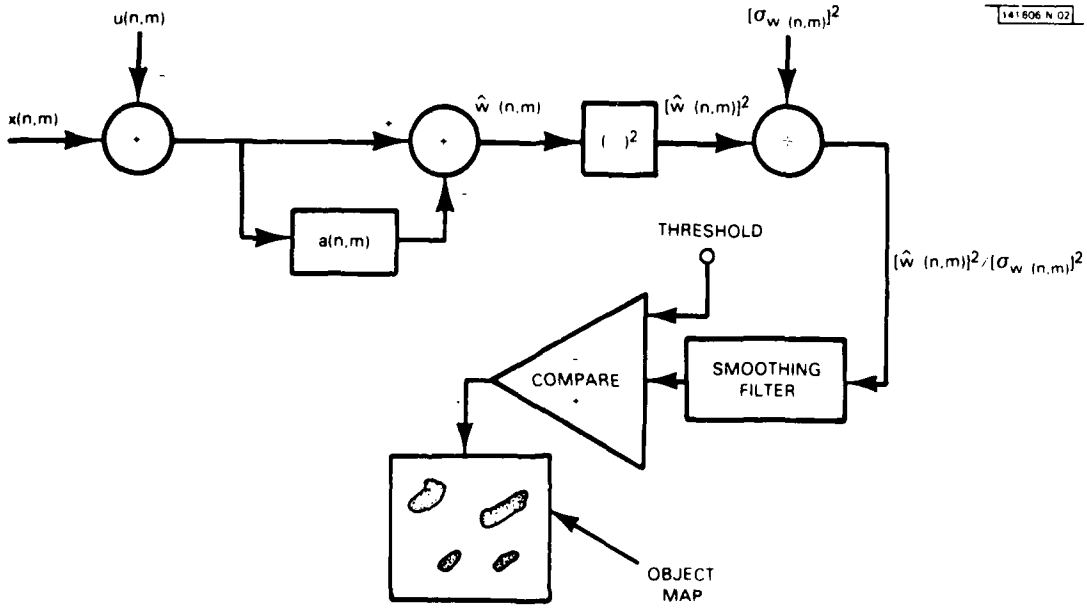


Figure 13. The detection algorithm.

The quantity on the right side of the inequality in (49) is a constant $T(P_F)$ to be determined.

The test (49) represents an exact equivalence to the original significance test (41) and requires a set of NM growing predictors per image pixel. A more practical approach utilizes the approximate whitening transformation based on fixed-order predictors (19). This transformation leads to (21) and thus to the approximate CFAR detector:

$$\text{"If } \sum_{\chi} \frac{(\hat{w}_{n,m})^2}{(\sigma_{w_{n,m}})^2} > T(P_F) \text{ , decide object present"} \qquad (50)$$

where $\hat{w}_{n,m}$ denotes the prediction residual computed from the observed data vector $\underline{\hat{x}}$. This decision requires only one (space-variant) predictor per image pixel. Equation (50) lends itself to a simple intuitive interpretation. Specifically, in generating the elements $\hat{w}_{n,m}$ we attempt to first whiten the data with a fixed-order space-varying inverse filter. Normalization by the variance $(\sigma_{w_{n,m}})^2$ gives equal weight to each residual. When anomalies are present, the values of the normalized residuals will increase, resulting in a higher likelihood of the statistic crossing the threshold $T(P_F)$. The approximate CFAR detection algorithm is depicted in Figure 13 where the indices (n,m) refer to the coordinates of the entire image x(n,m) and not the the sliding region χ. Before prediction and normalization, the local mean is first subtracted from the image. The final smoothing operation performs the summation in (50).

5.3 Adaptive Estimation

To apply the approximate signifiance test (50) to an image we must first estimate the autoregressive model coefficients, the local mean, and the prediction error variance at each image pixel location (n_o,m_o). The estimated parameters are used to generate the normalized prediction error residual for just the single point (n_o, m_o). The procedure is repeated at each point in the image to generate the terms in the sum in (50).

To estimate the parameters at each point we proceed as follows. Suppose that these parameters evolve slowly over the image. Under this condition, it is assumed that the image is stationary over a "sufficiently large" region to obtain a reliable estimate of the model parameters. For estimates at location (n_o,m_o), this region is assumed square of size BxB and centered[3] at (n_o,m_o) as illustrated in Figure 14. We refer to this BxB sliding window as the "estimation window". The support of this window is distinctly different from that of the decision region χ. In particular, the estimation window will generally be larger than χ, and extend as far as possible about its center while preserving approximate stationarity. Increasing the size of the estimation window decreases the variance of the parameter estimates and leads to a better estimate of the parameters since anomalies will have a smaller effect in a larger region.

To estimate the mean image level $u(n_o,m_o)$ we simply average the values under the estimation window centered at (n_o,m_o). To estimate the

[3]We have assumed that B is odd.

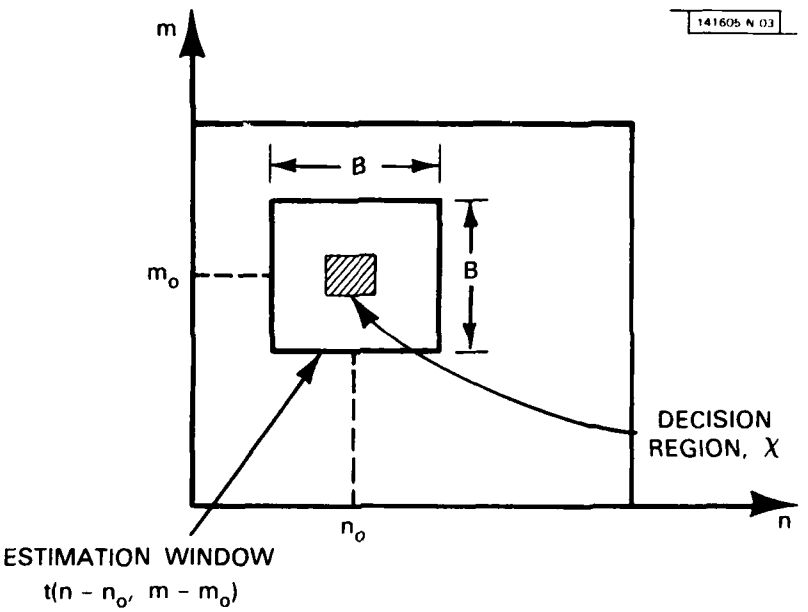


Figure 14. Representation of the estimation and decision regions.

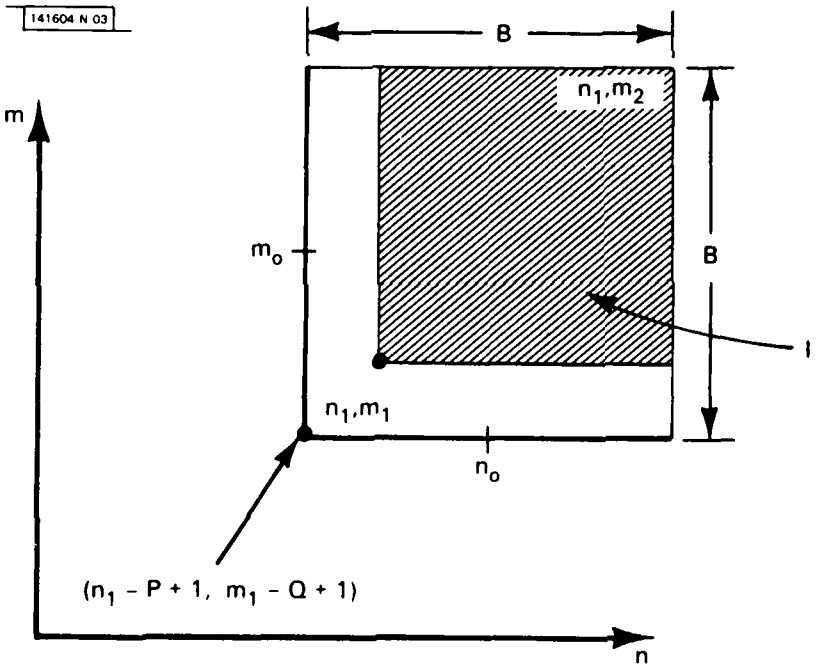


Figure 15. Data block used in 2-D least squares.

predictor coefficients $a(n_o,m_o;\ell,k)$ and prediction error variance $(\sigma_{w(n_o,m_o)})^2$ in (50), we use the "covariance method". We further assume for simplicity that the support of the coefficients is a PxQ first-quadrant mask. The following procedures, however, are clearly applicable to more general mask shapes. Our goal then is to estimate the model parameters $a(n_o,m_o;\ell,k)$ for $\ell=0,1,2,\ldots,P-1$ and $k=0,1,2,\ldots Q-1$ with $(\ell,k) \neq (0,0)$ and the corresponding prediction error variance $(\sigma_{w(n_o,m_o)})^2$ from the data under the estimation window. We first define the prediction error over the estimation window such that only the data <u>within</u> this region is used in prediction:

$$e(n,m) = x(n,m) - \sum_{\substack{j=0 \\ (j,k)\neq(0,0)}}^{P-1} \sum_{k=0}^{Q-1} a(j,k;n_o,m_o)x(n-j,m-k),\ (n,m)\in I \qquad (51)$$

where I denotes the region $[n_1,\ n_2]x[m_1,\ m_2]$ in which data elements are predicted, as illustrated in Figure 15. We then wish to minimize the sum of the squared errors given by

$$E[n_o,\ m_o] = \sum_{n=n_1}^{n_2} \sum_{m=m_1}^{m_2} e^2(n,m) \qquad (52)$$

To proceed with the minimization, we define the vector $\underline{a}[n_o,m_o]$ of length PxQ as the concatenation of the columns (or rows) of the unknown coefficient set $a(n_o,\ m_o;\ \ell,k)$, where we have assumed that $a(n_o,\ m_o;\ 0,0)=1.0$. Minimization of (52) leads to a matrix form of the Normal equations in (8):

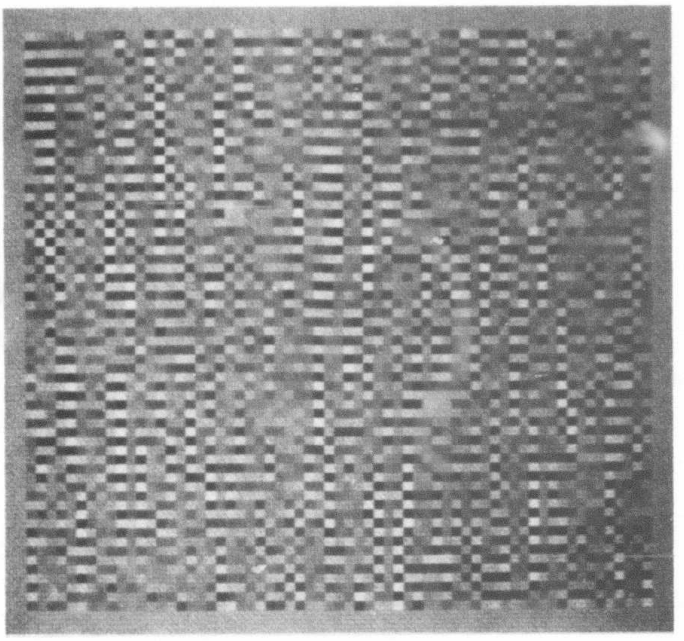
$$\hat{K}_x(n_o,m_o)\ \underline{\hat{a}}[n_o,m_o] = \begin{bmatrix} (\hat{\sigma}_{w(n_o,m_o)})^2 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \qquad (53)$$

where the matrix $\hat{K}_x(n_o,m_o)$ is an estimate of the covariance matrix K_x at (n_o,m_o), consisting of covariance estimates without assumptions about the observed data outside the estimation window. The terms $\underline{\hat{a}}[n_o,m_o]$ and $(\hat{\sigma}_{w(n_o,m_o)})^2$ represent estimates of the prediction coefficients and error variance, respectively. The last PxQ-1 equations in (53) can be used to solve for the elements $\underline{a}[n_o,m_o]$. This estimate of the model coefficients can then be used in the first equation of (53) to obtain an estimate of the prediction error variance.

5.4 Examples

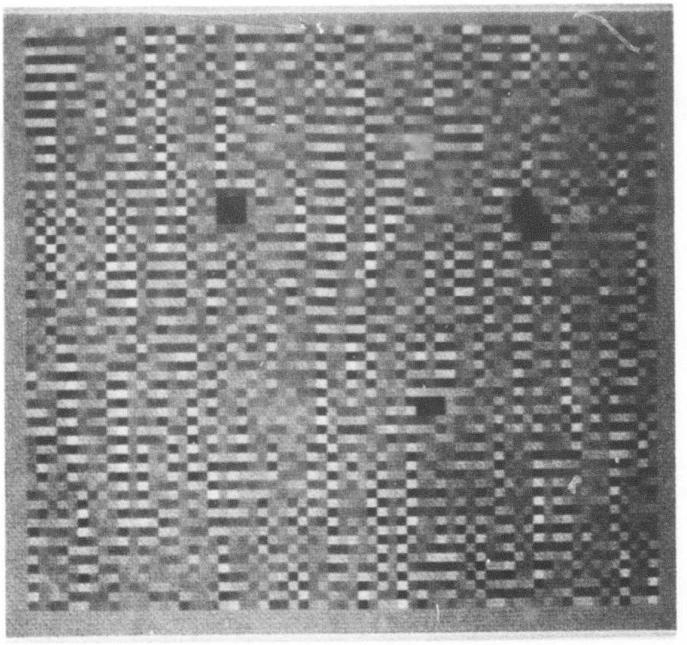
In this section, we present a number of examples of object detection in images. In these examples, the size of the estimation window was chosen to be 10x10 pixels, which we assume is "sufficiently" larger than the size of most objects and large enough to obtain accurate parameter estimates, while maintaining approximate stationarity. In addition, we have chosen a quarter-plane 2x2 pixel coefficient mask support. The adequacy of this support was demonstrated by observing little or no improvement with larger NSHP supports. Finally, a decision region of 3x3 pixels was chosen as

(a)



CP250-1017

(b)



CP250-1018

Figure 16. Demonstration of detection with synthetic data in example 1: (a) Test image with four objects, (b) Normalized prediction error with summing and thresholding.

reasonable for detecting objects of this size normally appearing in the processed data.

Example 1:

To demonstrate the detection algorithm under ideal conditions, we begin with a synthetic image generated by the 2-D difference equation of the form:

$$x(n,m) = 0.1x(n,m-1) - 0.9x(n-1,m) + 0.1x(n-1,m-1) + w(n,m) \quad (54)$$

Four objects are imbedded in this image as shown in Fig. 16a. These objects are of a constant level, close to the background mean, so they are visually difficult to detect.

Figure 17 shows the square of the prediction error $\hat{w}(n_o, m_o)$. The position of all four objects is clearly identified by the peaks in the prediction error. Fig. 16b shows the normalized prediction error summed over the decision region and thresholded. This is the result of the object detection algorithm. Note that all objects except the two most closely spaced are resolved. The two closely spaced objects are unresolved because their presence degrades the estimation of the variance at these points. For this data, which is known a priori to be stationary, the problem can be resolved by using a larger estimation window or by using one variance estimate for the entire image [4]. The later is illustrated in Fig. 16c where all four objects are resolved. In general, however, it will not always be possible to use a very large estimation window or a fixed background variance.

Example 2:

The photographic image displayed in Fig. 18a is of size 128x128 pixels. Figures 18a-18d depict the prediction error variance, the prediction error

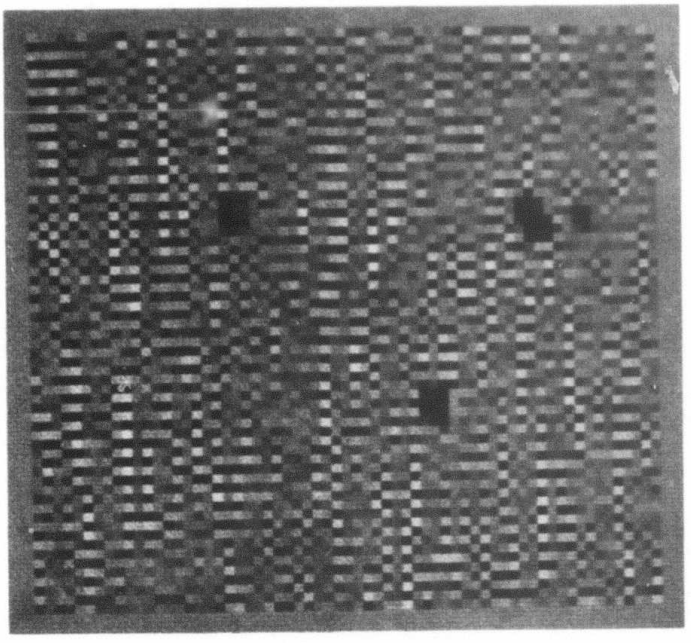


Figure 16c. Demonstration of detection with synthetic data in example 1, same as (b) with constant background variance.

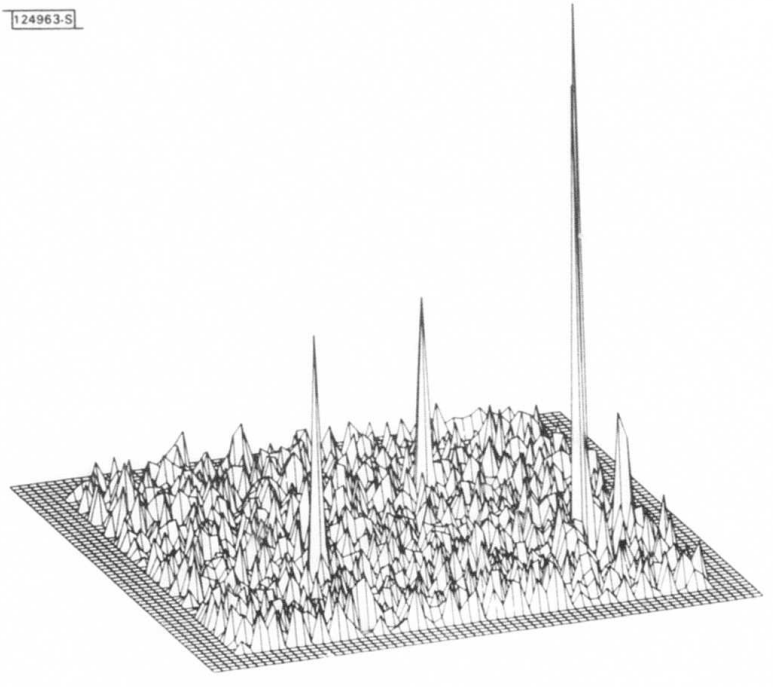


Figure 17. 3-D perspective of prediction error in example 1.

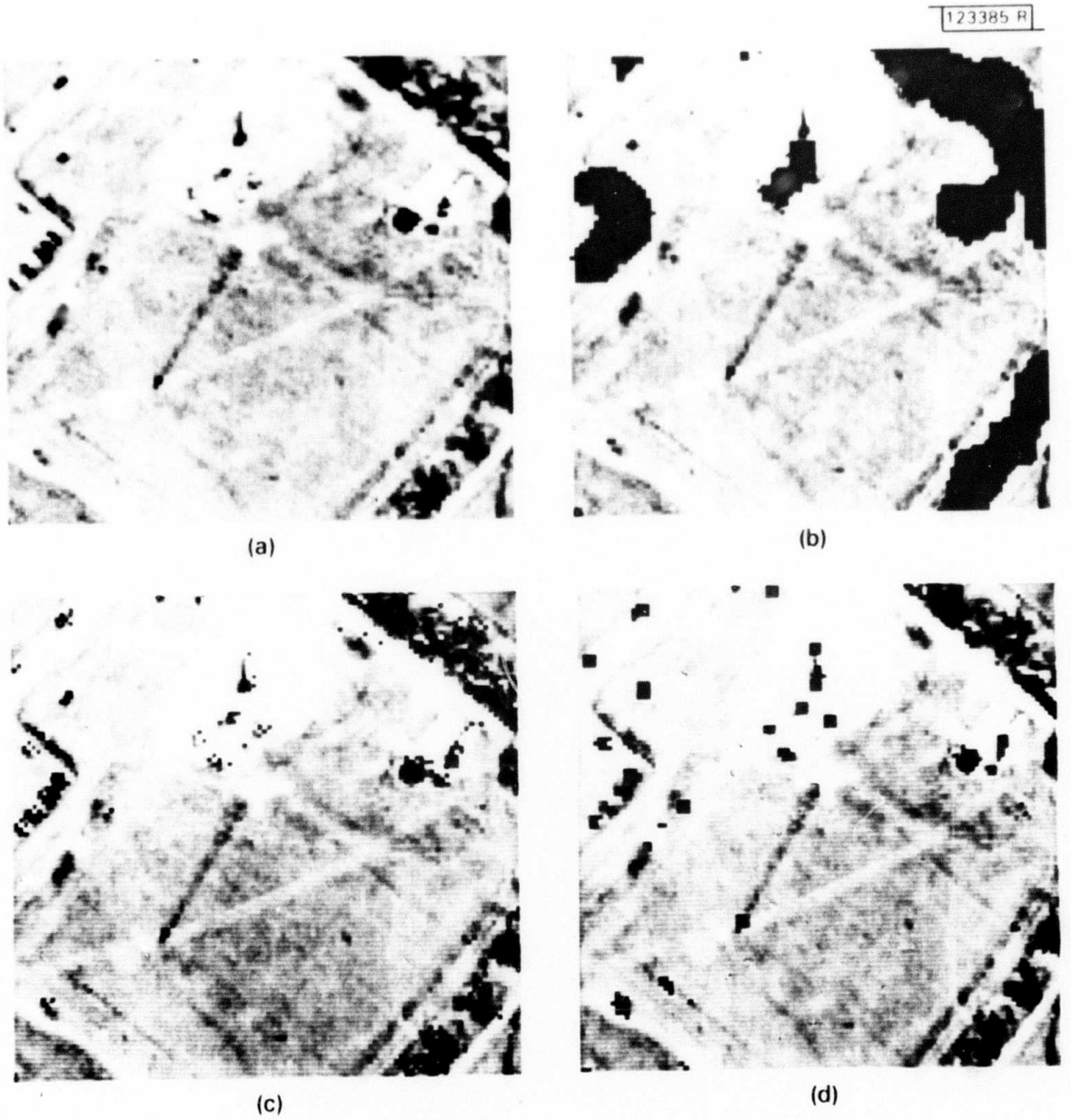


Figure 18. Demonstration of detection on real image data in example 2: (a) Photographic image, (b) Prediction error variance with thresholding, (c) Prediction error with thresholding, (d) Normalized prediction error with summing and thresholding.

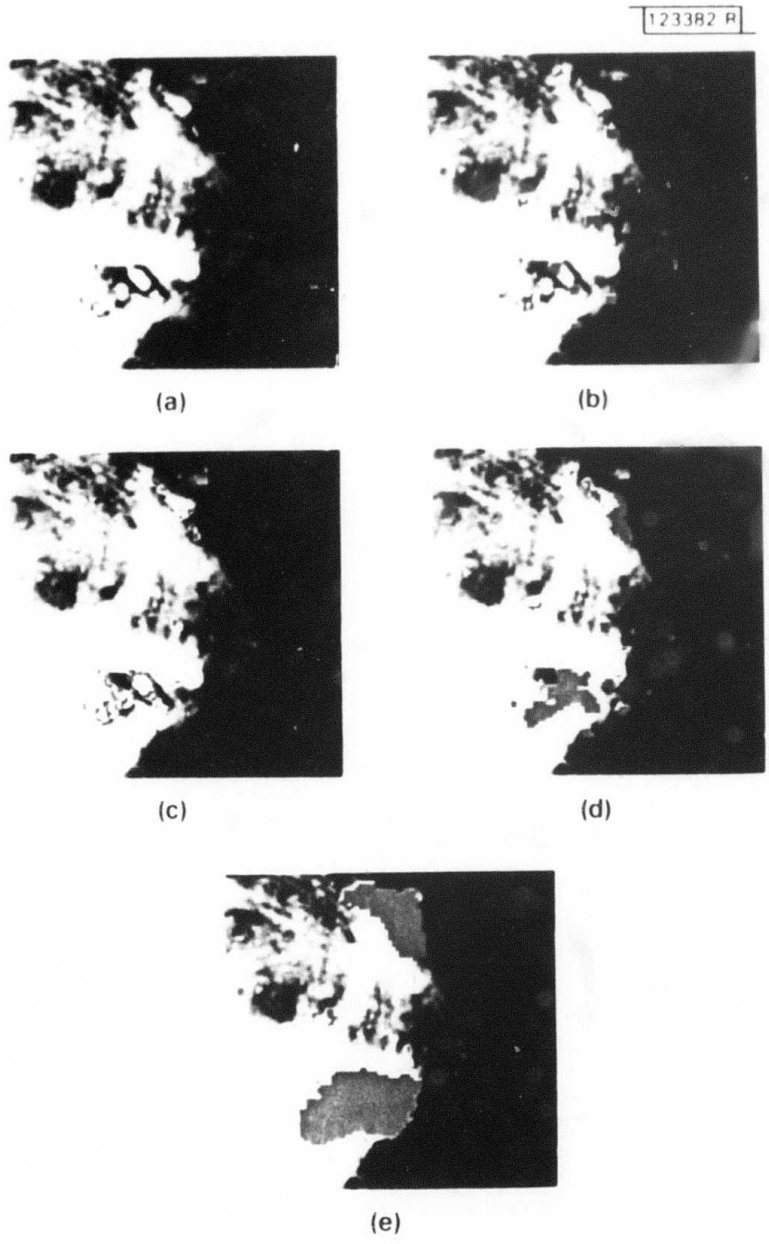


Figure 19. Demonstration of detection on real image data in example 3: (a) Photographic image, (b) Normalized prediction error with summing and thresholding, (c) Prediction error with thresholding, (d) Prediction error with summing and thresholding, (e) Prediction error variance with thresholding.

and the normalized prediction error summed over the decision region after thresholding. The algorithm has detected most objects in the different textured regions. A number of false alarms are observed in regions where background statistics vary rapidly.

Example 3:

The photographic image displayed in Figure 19a is of size 128x128 pixels. Figures 19b-19d depict the prediction error, and the prediction error summed over the decision region with and without normalization. In this case, the prediction error without normalization gives the most accurate indication of the presence of objects. As in example 1, this occurs because objects can introduce a false increase in the local background prediction variance, as illustrated in Figure 19e. Here the objects take up a sizeable portion of the estimation window and thus corrupt estimation of the background statistics.

A characteristic of this detection algorithm is that boundaries between regions do not show up as anomalous areas. Note that although the linear models do not apply at the boundaries between regions this seems to pose no problem. In particular, since the data in the estimation window is split at boundaries, we can expect that the estimated model parameters represent a compromise between the model parameters for each region. The corresponding prediction error is moderate throughout the regions near boundaries. The variance estimate however is large at a boundary since two different types of

data are present in the estimation window. Therefore, the normalized prediction error is low.

6. COMPUTATIONAL STRUCTURES

Having described a set of models and some image processing algorithms based on these models we now turn to the topic of computational structures to support these algorithms. In particular, we will look more closely at the computations involved in the target detection algorithm and describe the architecture of a special purpose machine based on systolic arrays to implement this algorithm. Since machine architecture is not the main topic of this paper, the discussion will be brief. However, we hope that our treatment will be sufficiently descriptive to convey some of the excitement we have for this particular area.

6.1 Overview

We begin by reviewing the steps in the target detection algorithm and detail the mathematical computations involved. We assume here a simpler algorithm that does not average the squared prediction error over the region χ. The image is scanned along rows and the linear prediction operation is repeated at each pixel as shown in Fig. 20.

The computational steps of the algorithm can be summarized as follows. First, a 2-D covariance matrix is formed from data in the estimation window. Next, a system of Normal equations (53) is solved to obtain the prediction filter coefficients $\underline{a}[n_o,m_o]$ and the prediction error variance $(\sigma_{w(n_o,m_o)})^2$. Finally, the prediction error is computed by applying the filter to the data point (n_o, m_o) at the center of the window and is

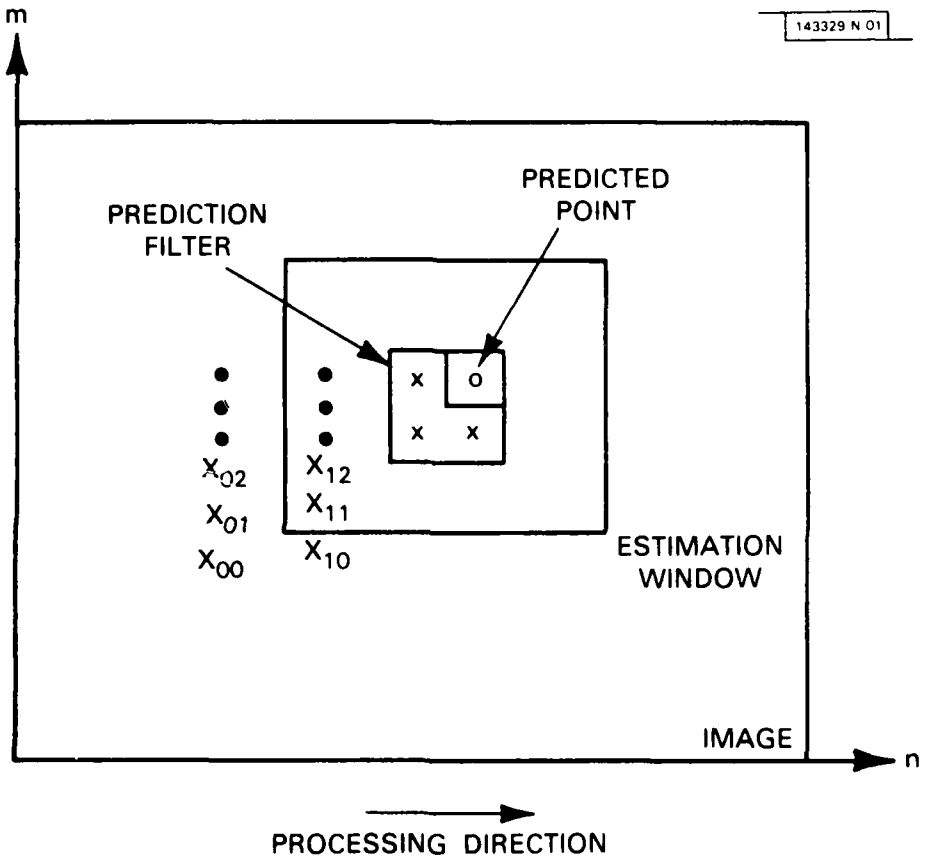


Figure 20. Linear prediction for object detection.

normalized by $\sigma_{w(n_o,m_o)}$. This normalized prediction error is then compared to a threshold to perform the target detection.

The steps just described can be implemented as a connected set of systolic arrays as shown in Figure 21. Data enters the lower array and elements of the covariance matrix are computed. As these elements are computed, they flow into the next array which begins computation of the filter coefficients. Finally, the filter coefficients are applied to the delayed input data to compute the prediction error. Details of the algorithm and array structures are given in the next two sections.

6.2 Computational Aspects of Target Detection Algorithms

In the following, let it be assumed that the linear predictive filter is 2x2 pixels and the estimation window is 8x8 pixels in size. These parameter values were found to be suitable for processing of typical aerial photographs. The image may be of any size and processing is assumed to proceed along rows.

Computation of the covariance matrix requires data within the estimation window and in one row above and one column to the left. Some of the required data points are labeled in Figure 20. For reasons that will become clear shortly, we shall be interested in forming the reversed covariance matrix $\tilde{K}_x$. This matrix is formally obtained from K_x by reversing the order of the elements in both the rows and column directions. The mean is assumed to be removed from the data prior to the following operations. This could be done in a separate pass through the image or simultaneously with computation

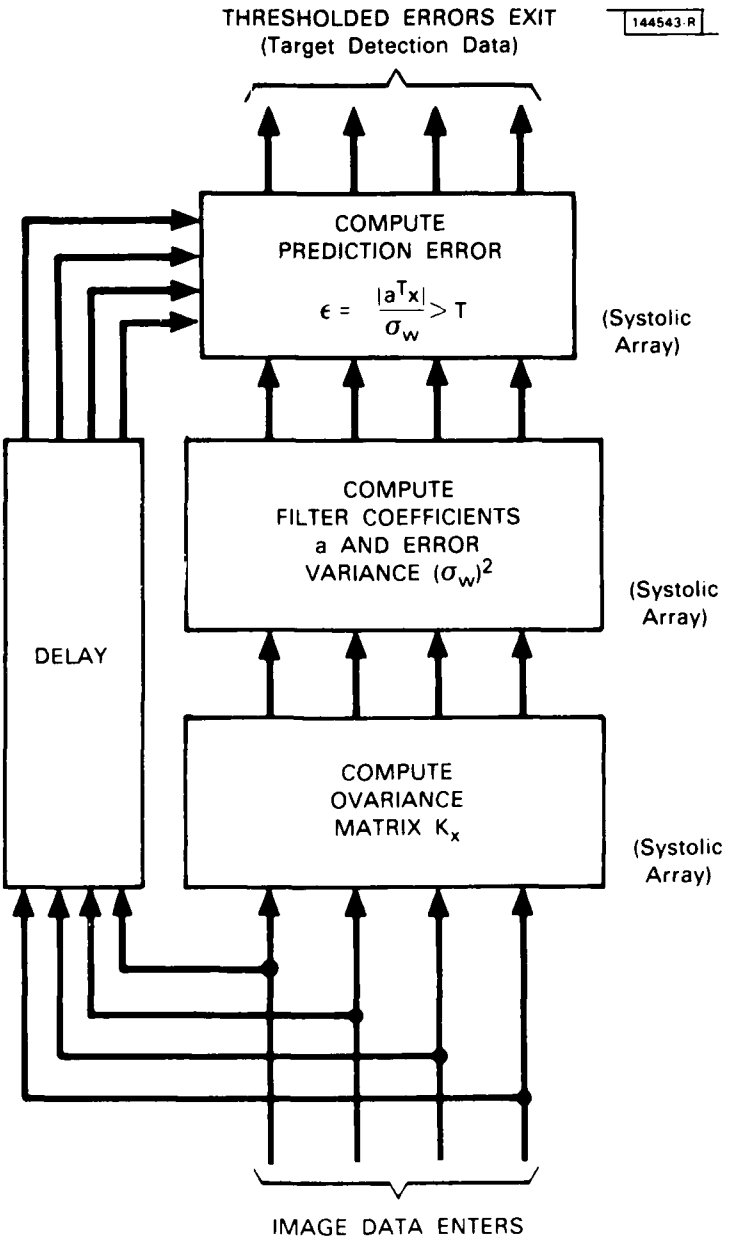


Figure 21. Flow of data and computations for object detection.

of the covariance [23]. If the covariance method of linear prediction is used, the reversed covariance matrix has the specific form

$$\tilde{K}_x = \frac{1}{B^2} F^T F$$

where

$$F = \begin{bmatrix} X_{00} & X_{01} & X_{10} & X_{11} \\ X_{01} & X_{02} & X_{11} & X_{12} \\ & \vdots & & \\ X_{07} & X_{08} & X_{17} & X_{18} \end{bmatrix} \} F_0$$

$$\begin{bmatrix} X_{10} & X_{11} & X_{20} & X_{21} \\ X_{11} & X_{12} & X_{21} & X_{22} \\ & \vdots & & \\ X_{17} & X_{18} & X_{27} & X_{28} \end{bmatrix} \} F_1 \qquad (55)$$

$$\vdots$$

$$\begin{bmatrix} X_{70} & X_{71} & X_{80} & X_{81} \\ X_{71} & X_{72} & X_{81} & X_{82} \\ & \vdots & & \\ X_{77} & X_{78} & X_{87} & X_{88} \end{bmatrix} \} F_7$$

and where B is the length and width of the estimation window.

By partitioning the matrix F as shown we can further write

$$\tilde{K}_x = \frac{1}{B^2} (F_0^T F_0 + F_1^T F_1 + \ldots + F_7^T F_7) \qquad (56)$$

This particular form is advantageous since seven of the terms remain in this sum when the computation is repeated at the point (n_o, m_o+1). Thus they

will not have to be recomputed. In addition, the reversed covariance matrix simple cells that perform the scalar operation $c \leftarrow c + a \cdot b$. A first array computes the matrix products $F_i^T F_i$ in succession. These terms are then fed into another array that computes a running sum of eight terms. In this way, the structure is able to compute one entire new covariance matrix as each column of data in the estimation window is scanned.

Once the covariance matrix is computed, the filter coefficients can be obtained by solving the Normal equations (53). If we use the reversed covariance matrix, these equations take the form

$$\frac{1}{B^2}\tilde{K}_x\tilde{\underline{a}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ (\sigma_w)^2 \end{bmatrix} \tag{57}$$

where for convenience we have dropped spatial dependence and where $\tilde{\underline{a}}$ is the vector of filter coefficients also in reversed order.

When written in this form the Normal equations can be solved in a manner that is especially easy to implement on a set of systolic arrays. In particular we have seen that the covariance matrix can be decomposed as in (22). Therefore, $\tilde{K}_x$ can always be written as the product LU where L is a lower triangular matrix with ones on the diagonal, and U is an upper triangular matrix. Consequently (57) can be rewritten as

$$U\tilde{\underline{a}} = L^{-1}\begin{bmatrix} 0 \\ 0 \\ 0 \\ B^2(\sigma_w)^2 \end{bmatrix} \tag{58}$$

(The last equality follows from the fact that L^{-1} is also lower triangular with ones on its diagonal.) Thus, if an LU decomposition is performed on the covariance matrix, a set of filter coefficients can be found by solving the triangular linear system. Fortunately, systolic array configurations exist both for performing the LU decomposition and for solving the triangular linear system (58). Thus these computations, like those needed to compute, K_x, can be pipelined through simple processors.

6.3 Systolic Array Processor Architecture

Figure 22 shows the architecture of the systolic array object detection processor. The computational methods are based on the early work of Kung and Lieserson [23] and use cells with characteristics described in Fig. 23. The cells in Fig. 23(a) perform a simple multiply/accumulate operation while those in Fig. 23(b) perform a division. Data flows through each cell, and data and computed results are available at neighboring cells for use during the next time increment (clock cycle). More details of the processor are given in Ref. 24 and in the general references on systolic arrays [23, 39-42]. The main hexagonal array in the lower left of Figure 22 computes one of the matrix product terms $F_i^T F_i$. The four linear arrays to the right of this array computes the running sum of eight such matrix products to form a complete covariance matrix. The terms of the covariance matrix flow into the upper hexagonal array which performs the LU decomposition which in turn feeds into a linear array for computing the coefficient. Finally, the coefficients are applied to the data points near

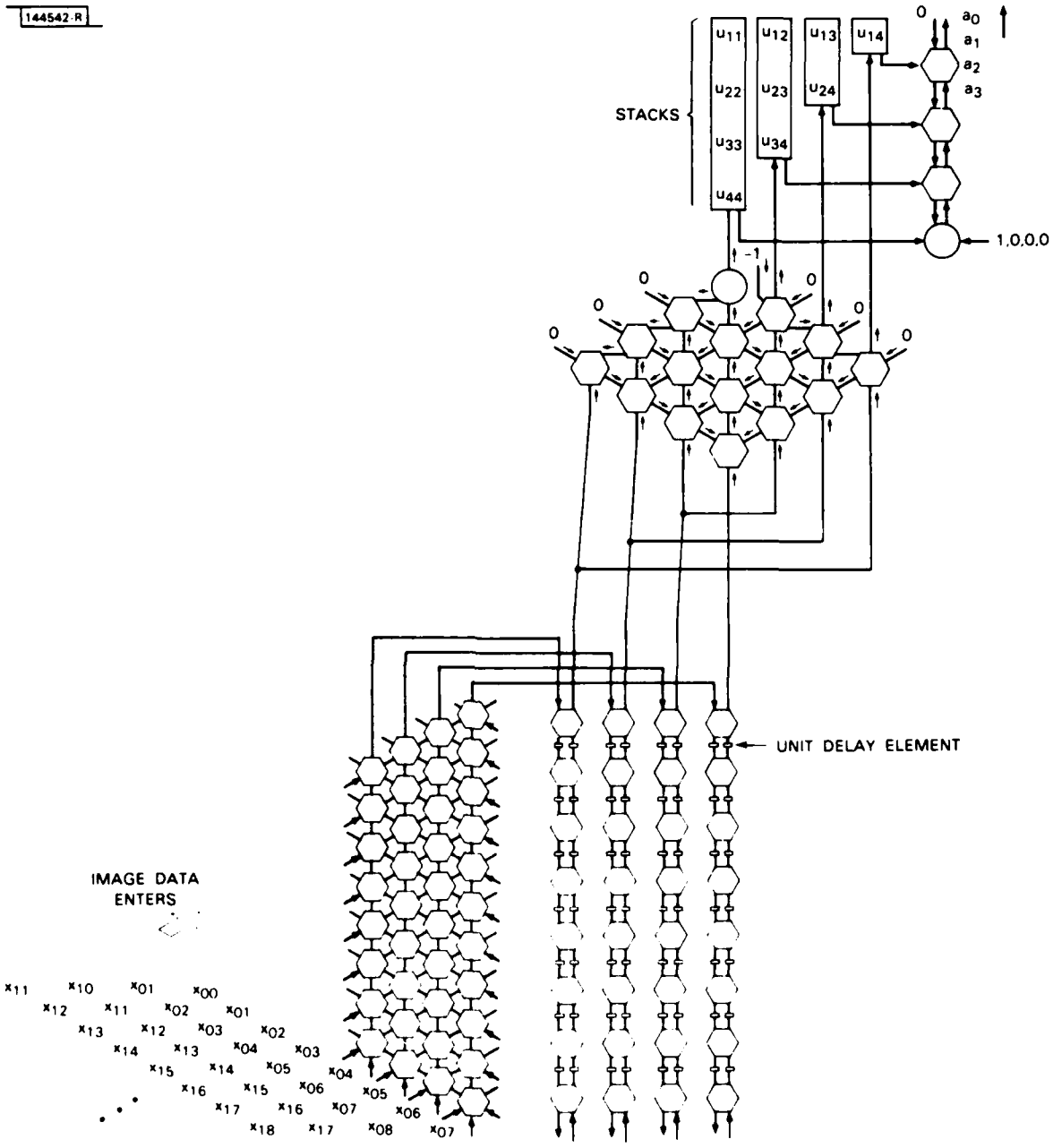
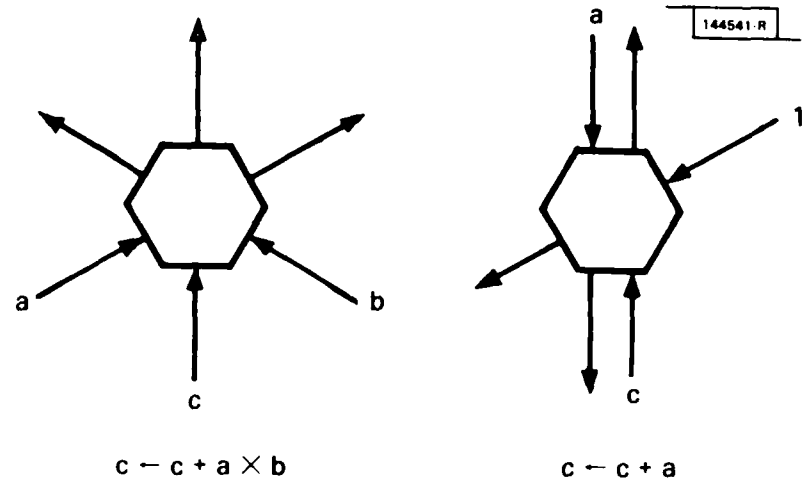
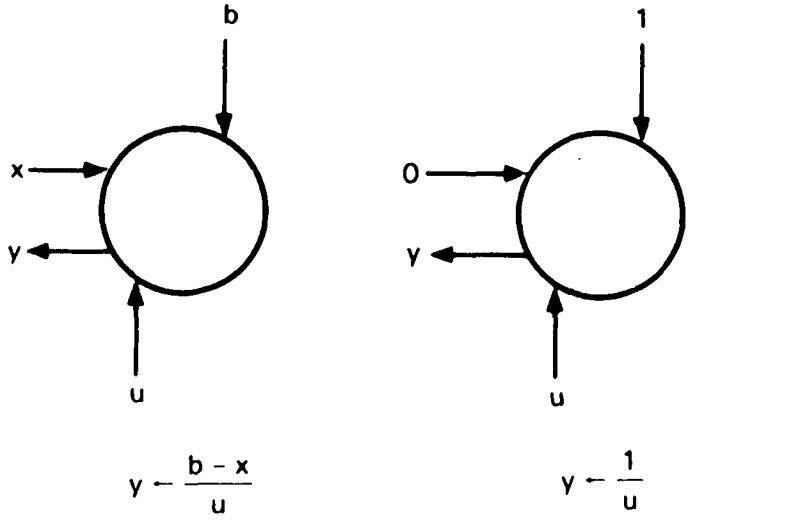


Figure 22. Systolic array object detection processor.



(a)



(b)

Figure 23. Cells used in systolic arrays. (a) Multiply-accumulate cells, (b) Special cells to perform division.

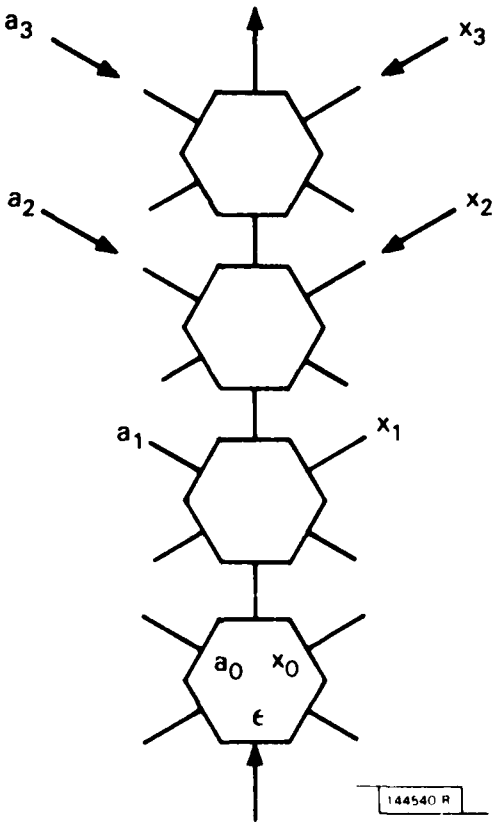


Figure 24. Systolic array for linear predictive filtering.

the center of the window to compute the prediction error residual. This last step is shown in Fig. 24.

The entire set of arrays work in lock step to convert input image data to target detection results. Currently, throughput is limited by the array that performs LU decomposition. Since this array has only 1/3 the throughput of the arrays that compute the covariance, the clock rate for the covariance array has to be slowed correspondingly.

The total initial delay for processing of a new column of image data is 154T where T is propagation time through a single cell. Thereafter, error residuals are computed at a rate of 12T. The initial delay can be represented as an "overhead rate" and is listed as a percentage of the steady-state processing time in Table I. Table I also gives the throughput of the processor for various sized images computed for T=1μs.

TABLE I
TARGET DETECTION ALGORITHM

PROCESSING RATES FOR VARIOUS IMAGE SIZES

Image Size	Overhead	Processing Rate*
64 x 64	22%	16.7 frames/sec
128 x 128	11	4.6 frames/sec
256 x 256	6	1/2 frames/sec
512 x 512	3	0.3 frames/sec
1024 x 1024	1	0.08 frames/sec

*Based on T = 1μs

6.4 Considerations for VLSI Implementation

The regular structure of the arrays and the use of a large number of simple cells should allow the entire processor for target detection to be implemented with emerging VLSI wafer scale technology. These considerations, in fact, led us to choose the present design for the processor over alternatives that use fewer, but considerably more complex cells (see e.g., Ref. 41).

The target detection processor described here requires 109 multiply/acummulate cells (Fig. 23a), two special purpose divide cells (Fig. 23b), buffer storage and delay. We believe these requirements could be met using the wafer scale restructurable VLSI technology currently under development at Lincoln Laboratory [43].[4] Restructurable VLSI allows working cells within a wafer to be utilized and defective cells to be bypassed in forming the necessary intercell connections.

[4] A similar system for performing 16-point FFT's at a data rate of 16 MHz has already been developed and tested [42]. this system employed 128 16-bit cells on a three inch wafer which were implemented with 5-μm, two-level metal CMOS technology.

7. CONCLUSIONS

In this report, two-dimensional stochastic linear models were used in developing algorithms for image segmentation and object detection. To accomplish this, we have relied on the merger of the stochastic representation of image textures, statistical inference procedures, and model identification and estimation. A common thread throughout our algorithms is the interpretation of the inference procedures in terms of linear prediction residuals. This interpretation leads to statistical tests more insightful than the original tests and makes the procedures computationally tractable. This computational efficiency was demonstrated with the object detection algorithm which served as a good example of an algorithm amenable to the special parallelism of systolic arrays.

More specifically, for the purpose of image segmentation this report developed a class of models for terrain images with two levels of structure. An underlying structure based on stochastic filtering concepts represents the texture in local regions of terrain. Superimposed on this structure is a Markov random field that describes transitions from one region type to another. Using those models, we considered segmentation as a region estimation problem and explored maximum likelihood and maximum a posteriori estimation procedures. The ML estimation ignores the Markov structure that describes the occurrence of regions and produces a "spotty" result. The MAP estimation utilizes the Markov structure but leads to a difficult optimization problem. A suboptimal solution can be obtained, however,

through a procedure that begins with the ML estimate and iterates to a final result. The region estimate thus produced is quite accurate. Examples of segmentation were shown for some aerial photographs of a rural area.

The detection algorithm developed in this report relies on a significance test which adapts itself to the changing background in such a way that a constant false alarm rate is maintained. This test has a potentially practical implementation, since it can be expressed in terms of the residuals of an adaptive two-dimensional linear predictor. In particular, the various steps of the algorithm of estimating a correlation matrix, solving the normal equations, and performing the prediction error filtering were shown to be amenable to systolic arrays most of whose cells perform the simple inner product $c \leftarrow c + a \cdot b$.

The approach developed in this report for object detection has recently been extended to accomplish other tasks in image analysis. In particular, the method of spatial linear prediction has been extended to detect region borders in aerial photographs [44, 45]. Such boundary detectors rely on a combination of various causal and non-causal linear predictors derived from an approximate significance test.

Although the object detection algorithm and its more recent extension were successfuly demonstrated on synthetic and real-world images, a number of areas need further investigation. For example, improved estimates of the model parameters and of background variance might be sought which avoid the corruptive influence of the presence of large objects or adjacent differing

textures. An iterative technique is one possibility. On each iteration, the background statistics might be estimated from pixels which do not include current object samples or unwanted textures. Such an approach may lead to more accurate object and boundary detection.

References

1. W.K. Pratt, Digital Image Processing (John Wiley and Sons, New York, 1978).

2. C.W. Therrien, "An Estimation-Theoretic Approach to Terrain Image Segmentation," Computer Vision, Graphics, and Image Processing, 22, 3, 313-326, (1983).

3. T.F. Quatieri, "Object Detection by Two-Dimensional Linear Prediction," Proceedings, Proceedings, IEEE Intl. Conf. Acoust., Speech, and Signal Processing, Boston, 14-16 (1983).

4. T.F. Quatieri, "Object Detection by Two-Dimensional Linear Prediction," Technical Report 632, Lincoln Laboratory, M.I.T., DTIC AD-A126340/9, (January, 1983).

5. B.H. McCormick and S.N. Jayaramamurthy,"Time Series Model for Texture Synthesis," Int. J. Comput. Inf. Sci., 3, (1974).

6. W.K. Pratt, O.D. Faugeras, and A. Gagalowicz, "Applications of Stochastic Texture Field Models to Image Processing," Proceedings, IEEE, 69, 542-551 (1981).

7. O.D. Faugeras, "Autoregressive Modeling with Conditional Expectations for Texture Synthesis," Proceedings, 5th Int. Conf. Pattern Recognition, Miami Beach, Fla., December 1-4, 1980.

8. W.K. Pratt and O.D. Faugeras, "Development and Evaluation of Stochastic-based Visual Texture Features," Proceedings, 4th International Conference on Pattern Recognition, 1978, Kyoto, Japan, November 7-10, 1978.

9. A. Gagalowicz, "Analysis of Texture Using a Stochastic Model," Proceedings, 4th International Conference on Pattern Recognition, Kyoto, Japan, November 7-10, 1978.

10. K.I. Laws, "Rapid Texture Identification," Proceedings, Soc. Photo-Opt. Instrum. Eng., SPIE 23, 376-380 (1980).

11. J.T. Tou and Y.S Chang, "An Approach to Texture Pattern Analysis and Recognition," Proceedings, IEEE Conference on Decision and Control, Clearwater, Fla., December (1976).

12. C.W. Therrien, "Linear Filtering Models for Texture Classification and Segmentation," Proceedings ., 5th International Conference on Pattern Recognition, Miami Beach, Fla., December 1-4, 1980.

13. C.H. Chen, private communication.

14. D.B. Cooper, et al, "Stochastic Boundary Estimation and Object Recognition," Computer Vision, Graphics, and Image Processing, 12, pp. 326-356, 1980.

15. D.B. Cooper, "Maximum Likelihood Estimation of Markov-Process Blob Boundaries in Noisy Images," IEEE Trans. Pattern Anal. Mach. Intelligence, PAMI-1, (1979).

16. H. Elliott, et al, "Implementation, Interpolation, and Analysis of a Suboptimal Boundary Finding Algorithm," Proceedings, IEEE Computer Society Conference on Pattern Recognition Image Processing, Chicago, Ill., August 6-8.

17. F.R. Hansen and H. Elliott, "Image Segmentation Using Simple Markov Field Models," Computer Vision, Graphics and Image Processing, 20, 101-132, (1982).

18. J.F. Claerbout, "Detection of P-Waves from Weak Sources at Great Distances," Geophysics, 24, 2, 197-211, (1964).

19. G.A. Clark and P.W. Rodgers, "Adaptive Prediction Applied to Seismic Event Detection," Proceedings, IEEE, 69, 9, 1166-1168, (1981).

20. C.D. Wang, "Adaptive Spatial/Temporal/Spectral Filters for Background Clutter Suppression and Target Detection," Optical Engineering, 21, 6, 1033-1038, (1982).

21. W.P. Dove and A.V. Oppenheim, "Event Location Using Recursive Least Squares Signal Processing," Proceedings, IEEE International Conf. Acoust., Speech, and Signal Processing, Denver, Colorado, April (1980).

22. D.T.L. Lee and M. Morf, "A Novel Innovative Approach Based Time-Domain Pitch Detector," Proceedings, IEEE International Conf. Acoust., Speech, and Signal Processing, Denver, Colorado, April 1980.

23. C. Mead and L. Conway, Introduction to VLSI Systems, Section 8.3, Addison-Wesley, Reading, MA, 1980.

24. C.W. Therrien, "A Systolic Array Structure for Target Detection," Proceedings, 17th Annual Asilomar Conference on Circuits, Systems, and Computers, Pacific Grove, CA, Oct 31 - Nov 2, 1983.

25. T.L. Marzetta, "Two-Dimensional Linear Prediction: Autocorrelation Arrays, Minimum-Phase Prediction Error Filters, and Reflection Coefficient Arrays," IEEE Trans. Acoust., Speech, Signal Process., ASSP-28, 726-733, (1980).

26. M.P. Ekstrom and J.W. Woods, "2-D Spectral Factorization with Applications in Recursive Digital Filtering," IEEE Trans. Acoustics, Speech, and Signal Processing, ASSP-24, 2, 115-178, April (1976).

27. D. Dudgeon and R. Mersereau, Multidimensional Digital Signal Processing, (Prentice Hall 1984).

28. H. Chang and J.K. Aggarwal, "Design of Two-Dimensional Semicausal Recursive Filters", IEEE Trans. Circuits and Systems, CAS-25, 12, 1051-1059, (1978).

28. A.K. Jain, "Advances in Mathematical Models for Image Processing," Proceedings, IEEE, 69, 5, May (1981).

29. M. Wax and T. Kailath, "Efficient Inversion of Toeplitz-Block Toeplitz Matrix", IEEE Trans. Acoust., Speech, Signal Process., ASSP-31, 5, 1218-1222, (1983).

30. A.V. Oppenheim and R.W. Schafer, Digital Signal Processing, (Prentice Hall 1975).

31. S.M. Kay and S.L. Marple, Jr., "Spectrum Analysis - A Modern Perspective", Proceedings, IEEE, 69, 11, 1380-1419, Nov. (1981).

32. J. Makhoul, "Linear Prediction: A Tutorial Review", Proceedings, IEEE, 63, 561-580, Apr. 1975

33. H.L. Van Trees, Detection, Estimation, and Modulation Theory, Vol 1, (John Wiley and Sons, New York 1968).

34. Drake, Fundamentals of Applied Probability Theory, (McGraw Hill, 1967).

35. C.W. Therrien, "On the Relation between Triangular Matrix Decomposition and Linear Prediction," Proceedings, IEEE, 71, 12, 1459-1460, December 1983.

36. J. Besag, "Spatial Interaction and the Statistical Analysis of Lattice Systems," J. Royal Statis. Soc., Ser. B. 36, 192-236, (1974).

37. A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene Labeling by Relaxation Operations," IEEE Trans. Syst. Man Cyber. SMC-6, 420-433, 1976.

38. R.A. Hummel and S.W. Zucker, "On the Foundations of Relaxation Labeling Processes," Proceedings, 5th International Conference on Pattern Recognition, Miami, Fla., pp. 50-63, December 1-4, 1980,

39. U. Weiser and A. Davis, "A Wavefront Notational Tool for VLSI Array Design," Proceedings, CMU Conference on VLSI Systems and Computations, H.T. Kung, B. Sproull, and G. Steele, Editors, (Computer Science Press, 1981).

40. H.T. Kung and M.S. Lam, "Fault-Tolerant VLSI Systolic Arrays and Level Pipelining," Proceedings, SPIE, Society of Photo-Optical Instrumentation Engineers, (1983).

41. W.M. Gentleman and H.T. Kung, "Matrix Triangularization by Systolic Arrays," Proceedings, SPIE, 298, Real Time Signal Processig IV, Society of Photo-Optical Instrumentation Engineers, (1981).

42. S.L. Garverick and E.A. Pierce, "A Single Wafer 16-point 16 MHz FFT Processor," Proceedings, IEEE 1983 Custom Integrated Circuits Conf., Rochester, NY, pp. 23-25 May 1983.

43. J.I. Raffel, "A Demonstration of Very Large Area Integration Using Laser Restructuring," Proceedings, Proceedings, IEEE Intl. Symp, on Circuits and Systems, Newport Beach, May 1983.

44. M.D. Richards, C.W. Therrien, and J.S. Lim, "Image Segmentation Using Spatial Linear Prediction", Proceedings, IEEE Intl. Conf. Acoustics, Speech, and Signal Processing, San Diego, CA, March, 1984

45. J.E. Bevington and R.M. Mersereau, "A Maximum-Likelihood Approach to Image Segmentation by Texture", Proceedings, IEEE Intl. Conf. Acoustics, Speech, and Signal Processing, San Diego, CA, March, 1984

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
ESD-TR-84-045

2. GOVT ACCESSION NO.
AD-A149 225

3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*
Statistical Signal Models and Algorithms for Image Analysis

5. TYPE OF REPORT & PERIOD COVERED
Technical Report

6. PERFORMING ORG. REPORT NUMBER
Technical Report 705

7. AUTHOR(s)
Thomas F. Quatieri
Charles W. Therrien
Dan E. Dudgeon

8. CONTRACT OR GRANT NUMBER(s)
F19628-85-C-0002

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Lincoln Laboratory, M.I.T.
P.O. Box 73
Lexington, MA 02173-0073

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
ARPA Order 3345
Program Element Nos. 61101E, 62702F and 62708E
Project Nos. 4594, 4D30 and 3T10

11. CONTROLLING OFFICE NAME AND ADDRESS
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209
Air Force Systems Command, USAF
Andrews AFB
Washington, DC 20334

12. REPORT DATE
25 October 1984

13. NUMBER OF PAGES
92

14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*
Electronic Systems Division
Hanscom AFB, MA 01731

15. SECURITY CLASS. *(of this report)*
Unclassified

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES
None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

systolic array	maximum likelihood estimation	image classification
stochastic linear models	parallelism	image segmentation
statistical inference	linear prediction	object detection

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

In this report two-dimensional stochastic linear models are used in developing algorithms for image analysis such as classification, segmentation, and object detection in images characterized by textured backgrounds. These models generate two-dimensional random processes as outputs to which statistical inference procedures can naturally be applied. A common thread throughout our algorithms is the interpretation of the inference procedure in terms of linear prediction residuals. This interpretation leads to statistical tests more insightful than the original tests and makes the procedures computationally tractable. This report also examines a computational structure tailored to one of the algorithms. In particular, we describe a processor based on systolic arrays that realizes the object detection algorithm developed in the report.